

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
**“Програмне забезпечення системи зберігання даних на базі
флеш-масивів у мережі”**

Виконав здобувач вищої освіти
IV курсу, групи KI-22-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Литвин Є.Ю.
« ____ » _____ 2026 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
« ____ » _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 27 » лютого 2026 року

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

Литвин Єлизаветі Юріївні

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи зберігання даних на базі флеш-масивів у мережі

2. Керівник роботи Усик Павло Сергійович, доктор філософії (PhD)

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

затверджені наказом вищого навчального закладу № 133-02 від 27.02.2026 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 зберігання даних на базі флеш-масивів у мережі

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 « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Усік П.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Литвин Є.Ю.
(прізвище та ініціали)

АНОТАЦІЯ

Литвин Є.Ю. Програмне забезпечення системи зберігання даних на базі флеш-масивів у мережі. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи зберігання даних на базі флеш-масивів у мережі.

Метою розробки є програмне забезпечення системи зберігання даних на базі флеш-масивів у мережі.

Результат роботи – програмна реалізація системи зберігання даних на базі флеш-масивів у мережі.

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'ютерна інженерія, система зберігання даних, флеш-масив, комп'ютерна мережа

ABSTRACT

Lytvyn Ye.Yu. Software for a data storage system based on flash arrays in a network. 123 Computer Engineering. 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, which is intended for a data storage system based on flash arrays in a network.

The purpose of the development is software for a data storage system based on flash arrays in a network.

The result of the work is a software implementation of a data storage system based on flash arrays in a network.

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 engineering, data storage system, flash array, computer network

3MIST

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми	34
3.4 Розробка діаграми процесів.....	37
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	39
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	39
4.2 Захист розробленого програмного забезпечення.....	62
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	64
6 ОСНОВНІ ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72

					ВКРБ-123.26.0010.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи зберігання даних на базі флеш-масивів у мережі	Літ.	Аркуш	Аркушів
Розроб.		Литвин С.Ю.				Б		
Перев.		Усік П.С.					1	78
Н.контр.		Коваленко А.С.				ЦНТУ КІ-22-1		
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

СЗД	–	Системи зберігання даних
ЦОД	–	Центр обробки даних
ШІ	–	Штучний інтелект
NVMe	–	Протокол для жорстких дисків
RAID	–	Технологія дублювання даних
SAS	–	Протокол для жорстких дисків
SSA	–	Масив на твердотільних накопичувачах
SSD	–	Твердотільні накопичувачі, пристрої зберігання даних, флеш-накопичувачі
VDI	–	Віртуалізація серверів
VSI	–	Віртуалізація робочих місць

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Масив класу All-Flash, також називаний SSA або масивом на твердотільних накопичувачах, – це система зберігання даних, утримуюча кілька флеш-накопичувачів. Система зберігання класу All-Flash не містить деталей, що рухаються, за рахунок чого виділяється менше тепла, споживається менше енергії й потрібно менше часу на обслуговування. З погляду функціональності технології класу All-Flash на базі процесорів Intel®Xeon® забезпечують виняткову продуктивність. Вони дозволяють скоротити затримку й поліпшити аварійне відновлення, використовувати аналітику в реальному часі й прискорити передачу даних. Все це дає IT-фахівцям можливість переключитися на інші завдання. Масиви на флеш-дисках служать базою для бізнес-застосунків наступного покоління й становлять основу центра обробки даних на флеш-дисках.

Система зберігання на флеш-накопичувачах – це носій, призначений для надійного зберігання даних в електронному виді. Носій можна перепрограмувати, витерши з його дані. Флеш-технологія дала новий імпульс розвитку IT. Флеш-системи не витрачають час на пошук і в них не виникають затримки через обертання, тому їхня швидкість відгуку на порядок вище, ніж у звичайних жорстких дисків. Система зберігання на флеш-дисках істотно підвищує швидкодію при виконанні операцій зберігання (уведення-вивід) і є основою для розвитку технологій наступного покоління.

Гібридний масив на флеш-дисках – це сполучення модулів зберігання даних, у яких використовується як мінімум два типи носіїв. Один з них – система зберігання на флеш-дисках, а другий може являти собою один з декількох варіантів. Гібридні масиви на флеш-дисках використовуються організаціями, яким необхідно зберігати активні й неактивні дані на одній платформі, а також потрібен баланс між продуктивністю й економічністю. Використовуючи гібридні

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

масиви на флеш-дисках, замовники можуть вибрати ідеальне співвідношення продуктивності і ємності з урахуванням своїх вимог, щоб одержати максимальну віддачу від інвестицій. Важлива перевага таких масивів – можливість додавати флеш-диски або ємність у міру необхідності.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи зберігання даних на базі флеш-масивів у мережі.

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

- Огляд існуючих систем зберігання даних на базі флеш-масивів у мережі.
- Дослідження системи зберігання даних на базі флеш-масивів у мережі.
- Програмна реалізація системи зберігання даних на базі флеш-масивів у мережі.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі зберігання даних на базі флеш-масивів у мережі.

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

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система зберігання на флеш-дисках – це носій для зберігання даних, що забезпечує істотний приріст продуктивності в порівнянні із традиційними жорсткими дисками. Система зберігання на флеш-дисках дозволяє значно знизити вартість операцій вводу-виводу. Це дає значну економію витрат для більшості робочих навантажень підприємства й для розгортання віртуальних ЦОД, які зараз широко використовуються в корпоративних середовищах. Віртуалізація приводить до істотного збільшення щільності операцій введення-виводу на сервер. Часто флеш-технології допомагають упоратися із цією проблемою, дозволяючи при цьому підвищити продуктивність і знизити витрати.

Флеш-технологія – це будь-яке сховище, у якому використовується флеш-пам'ять. Флеш-пам'ять випускається в різних формфакторах, тому ви, імовірно, використовуєте її щодня. Флеш-технологія використовується у всіляких пристроях – від флеш-мікросхем у простих друкованих платах і друкованих платах телефонів до повністю інтегрованих « флеш-дисків корпоративного класу», де замість традиційного жорсткого диска використовується кілька мікросхем. По суті, вона застосовується всюди!

1.2 Область застосування

Флеш-накопичувачі SSD – це пристрої зберігання даних, які дозволяють створити високопродуктивну систему зберігання з малими затримками, більше високою продуктивністю й енергоспоживанням, а також більше низькими витратами на охолодження в порівнянні зі звичайними обертовими носіями. Твердотільні флеш-диски виконують операції введення-виводу в 10 раз швидше,

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

ніж жорсткі диски. Замовники використовують флеш-диски для поліпшення часу відгуку застосунків у середовищах віртуальних серверів і віртуальних робочих місць. Крім того, ці диски дозволяють одержувати аналітичні дані в реальному часі для підтримки прийняття рішень.

Системою зберігання на флеш-дисках називається будь-який пристрій, що може функціонувати як сховище. Система зберігання на флеш-дисках може являти собою простий USB-пристрій або повністю інтегрований масив зберігання на флеш-дисках. Твердотільний диск (SSD) – це інтегрований пристрій, що прийшло на зміну традиційним обертовим дискам і яке широко використовується в корпоративних масивах зберігання даних.

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

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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

З появою таких рішень, як VMware Virtual SAN і StarWind Virtual SAN, багато користувачів стали виявляти цікавість до сховищ на базі серверних систем, побудованим повністю на флеш-накопичувачах. Саме собою, це все працює дуже швидко, але коштує пристойних грошей. Крім того, підприємства використовують і окремі системи зберігання даних All-Flash, що дає відмінні результати по IOPS, але для багатьох поки це коштує космічних грошей.

Виділяються наступні основні групи:

– **Elastic all-flash arrays.** Це дискові масиви, які споконвічно були запроектовані під флеш-сховища. Ця безліч містить у собі такі моделі, як Dell EMC XtremIO, Kaminario, Nimbus Data і Pure Storage, а аналітична компанія DCIG також включає туди масиви HPE Nimble й Dell EMC Isilon. Головна характеристика цієї групи – це можливість горизонтального масштабування, що має на увазі робочий процес "and-forget-it" (тобто налаштував і забув).

– **Enterprise all-flash arrays.** Це масиви корпоративного класу, які, як правило, коштують досить чималих грошей. Ця група представлена масивами Dell EMC VMAX, HPE 3PAR, Huawei OceanStor, NetApp AFF і Western Digital Tegile. У цих моделях є як вертикальне (scale-up), так і горизонтальне (scale-out) масштабування, а самі масиви заточені під видачу максимальної продуктивності (1+ million IOPS). Саме головне – що ці масиви добре обробляють змішані навантаження (а не просто послідовне читання або запис) по уведенню-виводу в більших корпоративних оточеннях (Enterprise-адміни зрозуміють, чому це найважливіша характеристика).

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

– **General purpose all-flash arrays.** Ця множина представлена такими продуктами, як Dell EMC Unity, FUJITSU Storage ETERNUS AF Series, Hitachi Vantara VSP F series, Nexsan Unity і NEC Storage M Series. Це традиційні дискові масиви із двома контролерами, які були оновлені, щоб обслуговувати флеш-накопичувачі. Головна перевага цих систем – розвинені системи керування, тому що вони пройшли великий шлях еволюції.

– **High performance all-flash arrays.** Це окремий клас дорогих систем на базі флеш-пам'яті, які тільки приходять на ринок. Як приклад зараз можна привести СЗД E8 Storage, а пізніше цього року вийде система від Kaminario. У цьому класі кількісна міра продуктивності вже на порядок вище – це 10+ мільйонів IOPS, одержуваних за рахунок використання технології NVMe-oF на фронтенд-інтерфейсах для хостов. Відзначимо, що впровадження цих сховищ на практиці поки утруднено, через високу ціну й загальною неготовністю інфраструктури до обслуговування таких сховищ.

– **Utility all-flash arrays.** Ця група містить у собі такі СЗД, як серію сховищ HPE MSA, IBM FlashSystem 900, серію NetApp E-series і SanDisk Infiniflash. Це системи для організацій, яким необхідно посадити на дисковий масив тільки кілька застосунків, яким необхідний високий рівень продуктивності й надійності. Ці сховища продаються за привабливою ціною, але заточені під цілком певні завдання й не підходять для середовищ із більшим числом навантажень. Крім того, у цих СЗД засоби керування не дозволяють робити багато чого.

All Flash масив від QSAN

Тема All Flash систем розбурхує розуми як користувачів, що бажають одержати у своє розпорядження зверхшвидкі сховища, так і виробників систем зберігання, що прагнуть задовольнити потреби своїх клієнтів. Усе більше й більше гравців вступають у битву за цей ласий шматок пирога, пропонуючи свої рішення в області зберігання на базі твердотільних накопичувачів. Не залишився осторонь і тайваньський виробник систем зберігання даних QSAN Technology, що

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

відносно недавно ввійшов у цю гонку, випустивши свій продукт All Flash – XCubeFAS XF2026D.

Взагалі ідея використовувати СЗД QSAN у якості All Flash не нова. Зокрема, ми вже згадували у своєму попередньому огляді лінійки XCubeSAN можливість укомплектовувати ці сховища винятково SSD. Завдяки можливості використання накопичувачів сторонніх виробників, можна одержати цілком бюджетне рішення, що володіє при цьому непоганою продуктивністю. Але завжди хочеться з наявних можливостей одержати максимум. Тому QSAN доробив свої СЗД для найкращої віддачі при роботі з SSD і випустив флагманський продукт XCubeFAS XF2026D, постачивши його голосним гаслом “Flash for all” (флеш для всіх).

Даний масив All Flash дійсно можна охарактеризувати як максимально доступний з погляду вартості володіння, оскільки в ньому збережена фірмова особливість продуктів QSAN у вигляді підтримки накопичувачів сторонніх виробників. Дійсно, чи багато знайдеться на ринку вендорів, що дозволяють такі вільності? Звичайні СЗД, укомплектовані SSD, не береться до уваги. Мова все-таки про спеціалізовані рішення, націлених на використання тільки SSD.

На перший погляд: яка різниця? Важливий же підсумковий результат. Однак, заборона на використання HDD може більш повно розкрити потенціал твердотільних накопичувачів (все-таки принципи роботи SSD і HDD абсолютно різні) і надати ряд по-своєму унікальних можливостей, наприклад, розширений моніторинг фізичного стану SSD. Розглянутий XCubeFAS XF2026D саме і є таким сховищем: використання HDD у ньому повністю виключено.

З погляду «заліза» XCubeFAS XF2026D є повним аналогом самої потужної СЗД із лінійки QSAN – XCubeSAN XS5226D: корпус 2U на 26 накопичувачів, два активних контролери на базі процесора Intel Xeon Quad Core серії D-1500, 2 мідних порти 10Gb iSCSI на кожному контролері. Але, на відміну від свого «звичайного» побратима, XCubeFAS має збільшений до 32 ГБ обсяг пам'яті й модуль захисту кеша Cache-2-Flash у комплекті поставки. За іншими

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

характеристиками можна звернутися до нашого огляду лінійки XCubeSAN. Тут же відзначимо ключові особливості:

– Досить цікавий корпус 2U26, що додає гнучкості при побудові дискових конфігурацій за рахунок додаткових слотів для дисків у порівнянні з конкуруючими продуктами.

– До 8 портів Fibre Channel 16G або до 10 портів 10Gb iSCSI на контролер дозволить не тільки збільшити пропускну здатність каналів зв'язку, але й у ряді випадків обійтися без використання дорогих FC/10Gb комутаторів.

Розширюється XCubeFAS за допомогою полиць розширення XCubeDAS, що мають, так само як і основний пристрій, форм-фактор 2U26. Усього підтримується до 4 полиць або в підсумку до 130 SSD у рамках єдиної системи. Не самий вражаючий показник, але, чесно говорячи, при нинішніх темпах росту ємності окремих накопичувачів цілком достатньо для абсолютної більшості сфер застосування.

Різниця з лінійкою XCubeSAN тут криється в «софті». Прошивання системи повністю переписана з урахуванням використання всіх переваг твердотільних накопичувачів. У результаті велика увага приділена моніторингу стану й ресурсу запису SSD як визначальну схоронність всіх збережених даних. Про кожний накопичувач можна одержати як узагальнену інформацію (живий-здоровий/необхідна заміна), так і детальні дані по кількості записаних на нього даних (глибина запиту – аж до року). А по даним ресурсу використання із самого накопичувача можна самостійно настроїти пороги оповіщення про деградацію SSD.

Також є детальний аналіз продуктивності всіх томів з тією же глибиною перегляду до одного року по ключових параметрах Latency, IOPS, пропускну здатність із можливістю вивантаження результатів у табличний файл для зовнішнього аналізу.

Досить непогано реалізовані рутинні операції по створенню/сміні пулів/томів/таргетів за рахунок засобів автоматизації. Адже не секрет, що висока

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

продуктивність All Flash систем досягається за рахунок максимальної паралелізації потоків вводу/виводу, що вимагає створення множини томів, таргетів, шляхів і так далі. Отож, QSAN пропонує пакетне створення/сміна таких об'єктів.

Відносно організації сховища даних в XCubeFAS застосовуються широко використовувані в цей час пули, усередині яких SSD поєднуються в RAID групи. Крім стандартних рівнів RAID диски можуть поєднуватися в так звані групи RAID EE, у яких диски hot spare використовуються не як окремі накопичувачі, а як набір блоків, рівномірно розподілених між всіма учасниками групи. У результаті резервні SSD також беруть участь в операціях вводу/виводу, вносячи свій внесок у підвищення швидкодії системи в цілому.

Зі звичних для сучасної СЗД функцій з захисту даних відзначимо підтримку снапшотів, клонів і асинхронної реплікації через iSCSI (включаючи інтеграцію з VSS). Причому, реплікація настроюється досить просто: адміністраторові досить указати в інтерфейсі системи тім для реплікації й адреса з логіном/паролем віддаленої системи. Інше буде виконано автоматично, включаючи необхідні налаштування на віддаленому масиві. Незважаючи на простоту дій, є й розширені налаштування в галузі розкладу, шейпинга, відказостійкості й балансування навантаження між каналами.

Зрозуміло, чималий інтерес представляють тести продуктивності. Тим більше, що порівняння sXCubeSAN XS5226D прямо-таки напрошується саме собою, щоб довідатися, наскільки спеціалізований масив All Flash відрізняється від звичайної СЗД.

ETERNUS AF

Система зберігання даних Fujitsu ETERNUS AF забезпечує всі переваги флеш-масиву й підтримку всіх застосунків першого рівня. Зверхмалий час відгуку, що гарантує рекордну продуктивність, функції дзеркалювання й перемикання на резервну систему у випадку виникнення неполадок, вибіркоче використання можливостей дедуплікації й стиску, а легко автоматична функція,

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

що також налаштовується, «якість обслуговування», що дозволяє скоротити обсяг операцій адміністрування. Ця система зберігання сприяє перетворенню ЦОД і зменшенню витрат, часу й необхідності втручання співробітників.

Huawei OceanStor Dorado

Компанія Huawei у своєму портфелі систем зберігання OceanStor має й масиви на базі флеш-пам'яті. Вони називаються Dorado і представлені двома моделями: 5100 і 2100 G2.

Масив Huawei OceanStor Dorado2100 G2 базується винятково на твердотільних накопичувачах і використовує поліпшену технологію керування кеш-пам'яттю, а також спеціалізовані алгоритми висновку-виводу-вводу-виводу. Одним з переваг масиву є його висока надійність. Він комплектується резервуємими джерелами живлення, вентиляторами, контролерами й інтерфейсними модулями. Накопичувачі SSD і BBU продубльовані, що забезпечує додатковий захист від одиничних відмов. У масиві використовуються спеціальні функції балансування ступеня зношування, завдяки яким вдається продовжити термін служби накопичувачів. Середній час стабільної роботи пристрою становить 1200000 годин.

Цей масив відмінно підійде для роботи з базами даних, хмарами й ERP-системами. Для керування масивом використовується кроссплатформений засіб OceanStor ISM, що забезпечує повного функціонала для налаштування масиву. З недоліків можна назвати відсутність звичного веб-інтерфейсу.

EMC ExtremIO

Це рішення від компанії EMC базується на стандартних компонентах. Це полегшує наступність покоління й відкриває можливість використання в майбутньому нових накопичувачів і процесорів.

Системи EMC не мають єдиної крапки відмови, а також мають убудовані джерела безперебійного живлення й контролери active-active.

Завдяки архітектурі Scale_out при масштабуванні системи збільшується й продуктивність і ємність.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Система складається із блоків Xbrick, доступні конфігурації з різним числом вузлів, від 1 до 6. Для ефективної роботи використовується дедуплікація на глобальному рівні, кожний кластер зберігає унікальні дані системи.

IBM FlashSystem 840

Рішення від компанії IBM орієнтована на використання в центрах обробки даних. Масив характеризується максимальною продуктивністю й добре підходить для підтримки ресурсномістких застосунків, систем оперативної обробки транзакцій, віртуалізації настільних систем, хмар і інженерних обчислень.

Масиви підтримують одночасну роботу великої кількості користувачів і дозволяє зменшити час пакетної обробки. Завдяки технології доступу до даних IBM MicroLatency поліпшується ефективність IT-Інфраструктури. Спеціалізована технологія захисту даних забезпечує високий рівень безпеки, підтримується шифрування AES-XTS з використанням 256-розрядного ключа, при цьому не наноситься збитку продуктивності. Основні компоненти системи резервуються, підтримуються елементи з гарячою заміною.

HPE StoreServ 7450

Розробка компанії HPE базується на спеціалізованій архітектурі, що забезпечує високу відказостійкість, ефективність і мобільність. Масив забезпечує високу продуктивність на рівні до 900 000 операцій висновку-виводу-вводу-виводу. Затримка при цьому не перевищує 0,7 мс. Ціна масиву також є його достоїнством: all-flash масив доступний за ціною систем на базі традиційних накопичувачів.

Масив передбачає убудовану дедуплікацію, що дозволяє зменшити витрати, завдяки економії обсягу. Використання спеціалізованої мікросхеми ASIC дозволяє зняти зайве навантаження із процесора й збільшує продуктивність всієї системи.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Порівняння систем

Найвищі показники продуктивності демонструють системи HPE і IBM. Розробка Huawei виграє по цьому показнику в масиву EMC. При цьому потрібно відзначити, що зі збільшенням IOPS в XtremIO збільшується й затримки, тоді як в OceanStor Dorado2100 G2 вона залишається стабільною. При цьому масив Huawei пропонує кращу пропускну здатність. Масив Huawei відрізняється більшою гнучкістю в порівнянні з масивом IBM FlashSystem 840 і забезпечує більше зручне функціональне програмне забезпечення. Ще одним важливим фактором є ціна. Розробка Huawei коштує нижче, ніж всі розглянуті рішення конкурентів.

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 кожен 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.
- 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.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

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 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.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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.

– Неєфективна 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

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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 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.

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи зберігання даних на базі флеш-масивів у мережі.

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-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розглянемо технології системи зберігання даних на базі флеш-масивів у мережі на основі продуктів Pure Storage. У першому кварталі поточного року, якщо вірити аналітикам Wells Fargo, продажу флеш-масивів уперше перевершили продажу зовнішніх дискових систем зберігання. Компанія Pure Storage була у свій час одним з перших стартапів в області флеш-масивів і в цей час є законодавцем мод у цьому сегменті ринку. Оберт компанії перевершує 1 млрд доларів. На недавній конференції Pure Accelerate компанія анонсувала оновлену лінійку моделей з підтримкою NVMe. Трохи раніше Pure Storage разом з NVIDIA представила готову інфраструктуру для підтримки рішень ІІІ. Крім того, сама компанія використовує предиктивну аналітику Pure1 Meta із застосуванням методів машинного навчання для підвищення надійності масивів.

Зараз практично будь-яка компанія, будь те банк або телеком-оператор, фактично стає ІТ-компанією, що робить сервіс своїм клієнтам. Інакше кажучи, роль ІТ істотно розширилася, а в результаті вимоги до рішень, включаючи СЗД, вирости, адже кінцевим результатом роботи ІТ є забезпечення конкурентної переваги для бізнесу.

Як раніше виглядав ринок СЗД? Традиційно важкі виробничі завдання розміщалися на масивах старшого класу, при виборі яких ціна була далеко не визначальним фактором. Головне, щоб усе працював як годинник. Далі йшли більше легкі завдання, наприклад тести або розробка, для яких було досить менш надійних масивів середнього класу, тут ціна була звичайно ж на першому місці.

Згодом багато компаній звернули увагу на те, що вартість супроводу їхніх масивів старшого класу висока, при цьому вони не задовольняють потреби бізнесу як по надійності, так і по швидкості роботи. Те ж саме справедливо й

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

відносно систем середнього класу: виявилося, що найчастіше по порівнянній вартості можна знайти більше надійне рішення. У результаті в пошуках альтернативи компанії приходять в Pure Storage.

За результатами останнього фінансового року оберт Pure Storage склав більше 1 млрд доларів. З обліком того, що Pure Storage робить тільки флеш-масиви (All Flash Array, AFA), по обсязі продажів у цій категорії ми перевершили результати багатьох давно відомих виробників. А в загальному заліку СЗД це дозволило нам увійти в десятку найбільших учасників ринку, включаючи виробників дискових і гібридних масивів. Серед наших клієнтів – велика кількість компаній зі списку Fortune 500 з інсталяціями на десятки й сотні петабайт даних.

На перший погляд, може здатися, що AFA вибирають через їхню високу продуктивність. Але насправді замовникам з боку бізнесу взагалі однаково, на якій платформі будуть працювати їхнього завдання. Головне – забезпечення безперервності бізнесу й дотримання SLA. Далі слово за фінансистами, яких цікавить загальна вартість володіння (TCO).

Окремо тут можна виділити вартість апгрейдів і продовження гарантії. Погодьтеся, ніхто не хоче кожні три-п'ять років купувати новий масив, повторно оплачувати ту ж ємність і вплутуватися в проект по міграції даних, тим більше що такий проект – залежно від розміру компанії – може займати до напівроку. Мало того, міграція є ще й досить витратним заходом.

Одним із ключових відмінностей рішень Pure Storage є те, що в наших масивів фактично відсутній граничний строк служби (end of life), а будь-який масив, навіть самий старий, може бути модернізований до нового покоління без переривання обслуговування. Це досягається за рахунок архітектури, що дозволяє без зупинки міняти компоненти й забезпечувати зворотну сумісність із усіма попередніми поколіннями.

Ця особливість дає нам можливість запропонувати клієнтам бізнес-модель Evergreen Gold, у рамках якої при продовженні підтримки ми кожні три роки

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

модернізуємо масив до наступного покоління. Представте, що ви їдете на машині зі швидкістю 120 км/год і на ходу, не вивантажуючи багаж і не висаджуючи пасажирів, перетворюєте її в новинку що тільки вийшла.

Ще один важливий факт: масиви Pure Storage поставляються із завжди включеним функціоналом на повну ємність, що також дозволяє заощаджувати клієнтам надалі.

Тут повторюється історія з дисковими масивами. Щоб одержати всі переваги NVMe, необхідно розробити дизайн масиву з урахуванням нового протоколу. Як показує практика, переробка старої архітектури не допомагає, потрібно проектувати масив з нуля.

Травень традиційно є часом анонсів новинок на ринку СЗД. На конференції Pure Storage Accelerate у Сан-франциско було анонсовано наступне покоління лінійки AFA-масивів Pure Storage на базі NVMe, при цьому модельний ряд був істотно розширений. У величезного числа наших клієнтів, включаючи російських замовників, системи подібного класу використовуються для підтримки важких продуктивних навантажень уже більше року. Перші системи Pure Storage з NVMe почали поставлятися пари років тому. Сьогодні Pure знову лідирує зі своїми інноваційними рішеннями.

Що дивно, ряд великих гравців ринку продовжують жити у світі SSD у силу довгого циклу розробки, а також у зв'язку з тим, що перехід на нову технологію означає для них відмова від ряду продуктів і втрату частини бізнесу. Клієнтам же говориться, що NVMe їм не потрібний. Або другий варіант – купуються готові NVMe SSD і встановлюються замість шпиндельних дисків у дисковий масив не менш чим 20-літньої давнини. При цьому упускається той факт, що ПЗ відіграє найважливішу роль для швидкості роботи системи й просто заміни носіїв тут недостатньо. Знову повторюється історія з SSD, чи не так?

І знову підкреслю: без повної переробки архітектури (контролери, операційна система, внутрішня шина, флеш-модулі) перехід на NVMe не принесе ніяких вигід.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Ми віримо, що робота з нашим масивом повинна бути настільки ж проста, як і керування автомобілем з автопілотом. Від розроблювачів це вимагає більших зусиль. Кілька років назад ми створили своє озеро даних, у якому консолідується телеметрична інформація з масивів, і механізм Pure1, що дозволяє пророкувати виникнення технічних проблем на масиві. Це ключові елементи для роботи нашої «СЗД із автопілотом».

Першим пріоритетним завданням було усунення можливих проблем з масивом, і для цього нам потрібен був аналіз «відбитків пальців», по яких можна було виявити потенційну проблему. Дивитися, у вас є тисячі масивів, установлених по усьому світі й генеруючих великий обсяг статистики, яку можна використовувати (важливо відзначити, що ніяких користувальницьких даних там немає). По-перше, ми надаємо ці дані клієнтам, у тому числі через додатки на мобільних телефонах. По-друге, за аналогією з антивірусами, ми скануємо ці дані на предмет можливих проблем у реальному часі.

Це дозволяє зробити підтримку предиктивною і скоротити кількість виникаючих інцидентів.

Ми надали клієнтам можливість планувати завантаження масиву з обліком майбутніх апгрейдів. У першому наближенні оцінка продуктивності масиву представляється очевидної. Але ми намагаємося, щоб завантаження ЦПУ на наших флеш-масивах були завжди близькі до 100%. Потенційна проблема полягає в тім, чим зайнятий масив: обробкою уведення-виводу при навантаженні або подальшому стиску даних, які вже перебувають на масиві. А що, якщо ваше навантаження виросте, наприклад, на 20% у найближчому майбутньому?

Неможливість пророчити ситуацію «що-якщо» була б істотним стримуючим фактором для адміністраторів. І тут ми вирішили звернутися до нашому Pure1, у якому вже є телеметрія з великої кількості наших масивів за минулі періоди. Далі справа за машинним навчанням. Розбиваємо телеметричні дані на тренувальний і тестовий набори, проганяємо їх через різні навчальні

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

алгоритми й використовуємо для створення моделей, потім порівнюємо моделі з реальними результатами на масиві й вибираємо кращу.

Головною перевагою такого підходу є те, що він забезпечує високу точність за рахунок використання досвіду кожного із клієнтів, підключених до Pure1.

Попит на флеш-масиви зараз дуже великий, тому що вартість флеш-накопичувачів знижується й компанії, які декількома роками раніше придбали б системи старшого класу під основні завдання або системи середнього класу під менш критичні, зараз повністю будують свої ЦОД на флеш-масивах. Саме із цим зв'язаний такий значний ріст компанії Pure Storage.

Флеш-масиви використовуються для всіх основних завдань підприємства. Ключові з них – розміщення баз даних (як OLTP, так і аналітика), віртуалізація серверів (VSI) і робочих місць (VDI). Залежно від завдання на перший план виходять різні переваги. Для баз даних – це швидкість і час відгуку, для середовищ віртуалізації – дедуплікація й компресія, які забезпечують високу фінансову ефективність. Середній коефіцієнт стиску реальних користувальницьких даних по нашій базі становить 5:1. Що важливо, залежно від завдання ми письмово гарантуємо значення й у випадку, якщо воно не досягається, робимо допоставку ємності. Хто ще може запропонувати клієнтам подібне?

Серед наших клієнтів у світі є банки й телекому-провайдери, а також промислові компанії, що займають верхні позиції в рейтингах. Обсяг інсталяцій часто перевищує десятки петабайт на один замовника. Але найважливіше те, що серед клієнтів Pure Storage дуже багато передових компаній із Кремнієвої долини, які, як правило, купують тільки інноваційні рішення.

Ми віримо, що флеш-масиви Pure Storage є більше якісною альтернативою масивам старшого класу як по надійності, так і за часом відгуку, а з обліком дедуплікації, компресії й вигідній вартості володіння вони цілком можуть конкурувати за вартістю з масивами середнього класу. Тому ми рекомендуємо

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

нашим існуючим і майбутнім клієнтам розглядати Pure Storage у будь-якому проекті, що включає закупівлю СЗД.

Для бізнес-керівництва, що не бажає глибоко поринати в технічні деталі, аргументація дуже проста: на ринку України з'явився новий гравець, чотири роки підряд займає лідируючі позиції у звітах таких відомих аналітиків як Gartner і IDC, що вийшов на оберт 1 млрд доларів і впевнений ріст, що показує тривалий. Pure Storage пропонує замовникам інноваційні рішення, що володіють невисокою вартістю володіння і які забезпечують схоронність інвестицій.

3.2 Розробка структурної схеми

Найшвидші у світі сервери й мережні підключення не можуть забезпечити ефективність роботи, якщо швидкість читання або запису даних у системі зберігання даних (СЗД) низька. Робота з базами даних, проведення часу аналізу, що віднімає масу, даних, використання онлайн-застосунків, доступ до яких необхідно забезпечити відразу багатьом користувачам, а також підтримка віртуалізованих серверних середовищ або середовищ віртуальних робочих місць – для рішення всіх цих завдань необхідною умовою є висока продуктивність всієї системи, включаючи СЗД. У протилежному випадку, СЗД може швидко стати «вузьким місцем», що обмежує продуктивність всієї системи.

Твердотільні накопичувачі (SSD) можуть збільшити швидкість обміну даними в сотні разів, значно перевершуючи по продуктивності жорсткі диски. Якщо всі дані, використовувані додатками з високими вимогами до продуктивності, розмістити в СЗД на базі твердотільних накопичувачів, то всі проблеми продуктивності й швидкодії будуть негайно вирішені. Хоча твердотільні накопичувачі обійдуться дорожче, якщо дивитися тільки на ціну, додаткові витрати будуть компенсуватися, якщо застосовувати ці накопичувачі в середовищах, що пред'являють високі вимоги до продуктивності.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Тимчасові рішення на базі жорстких дисків також можуть бути дорогими, однак продуктивність таких систем набагато менше тої, котра забезпечується в системах на базі твердотільних накопичувачів. Такі підходи, як інвестиції у виділені СЗД для конкретного застосування, розподіл даних по максимальній кількості жорстких дисків або часте налаштування системи є в найкращому разі тимчасовими рішеннями. А їхнє впровадження звичайно веде до недостатньо ефективного використання СЗД у цілому.

Швидкість – краща зброя, що дозволяють вижити й процвітати в умовах твердої конкуренції. На сьогоднішній день найвищу швидкість для досягнення ваших цілей пропонує протокол NVMe.

Традиційний протокол SAS, розроблений для жорстких дисків, не дозволяє твердотільним накопичувачам розкрити свій потенціал через складну архітектуру системи, занадто великої кількості протоколів для обробки й обмеженого числа паралельних черг. Компанія NVM Express сформулювала стандарти протоколу NVMe і замінила складну структуру протоколів у системі SAS, таких як планувальник вводу/виводу й SCSI, оптимізованим протоколом NVMe. У результаті низькі затримки й відмінна продуктивність при роботі із флеш-масивами (all-flash arrays, AFA) зробили NVMe найбільш підходящим вибором для великих підприємств.

Детально пророблена підтримка NVMe, включаючи комплексну розробку флеш-контролерів NVMe, ОС для флеш-масивів NVMe і твердотільних накопичувачів NVMe. Слід зазначити, що архітектура NVMe і алгоритми узгодження взаємодії диск – контролер забезпечують для флеш-накопичувачів на базі NVMe стабільну затримку на рівні всього 0,5 мс.

Флеш-масиви на базі NVMe перевершують аналогічні системи на базі SAS, але чому?

По-перше, на рівні передачі у флеш-масивах на базі SAS дані вводу/виводу передаються від процесора на накопичувач у такий спосіб:

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Крок 1: Дані вводу/виводу передаються від процесорів до чипів SAS по каналах і комутаторам PCIe.
- Крок 2: Перед доставкою на накопичувач SSD дані вводу/виводу перетворюються в пакети SAS за допомогою відповідних чипів SAS.

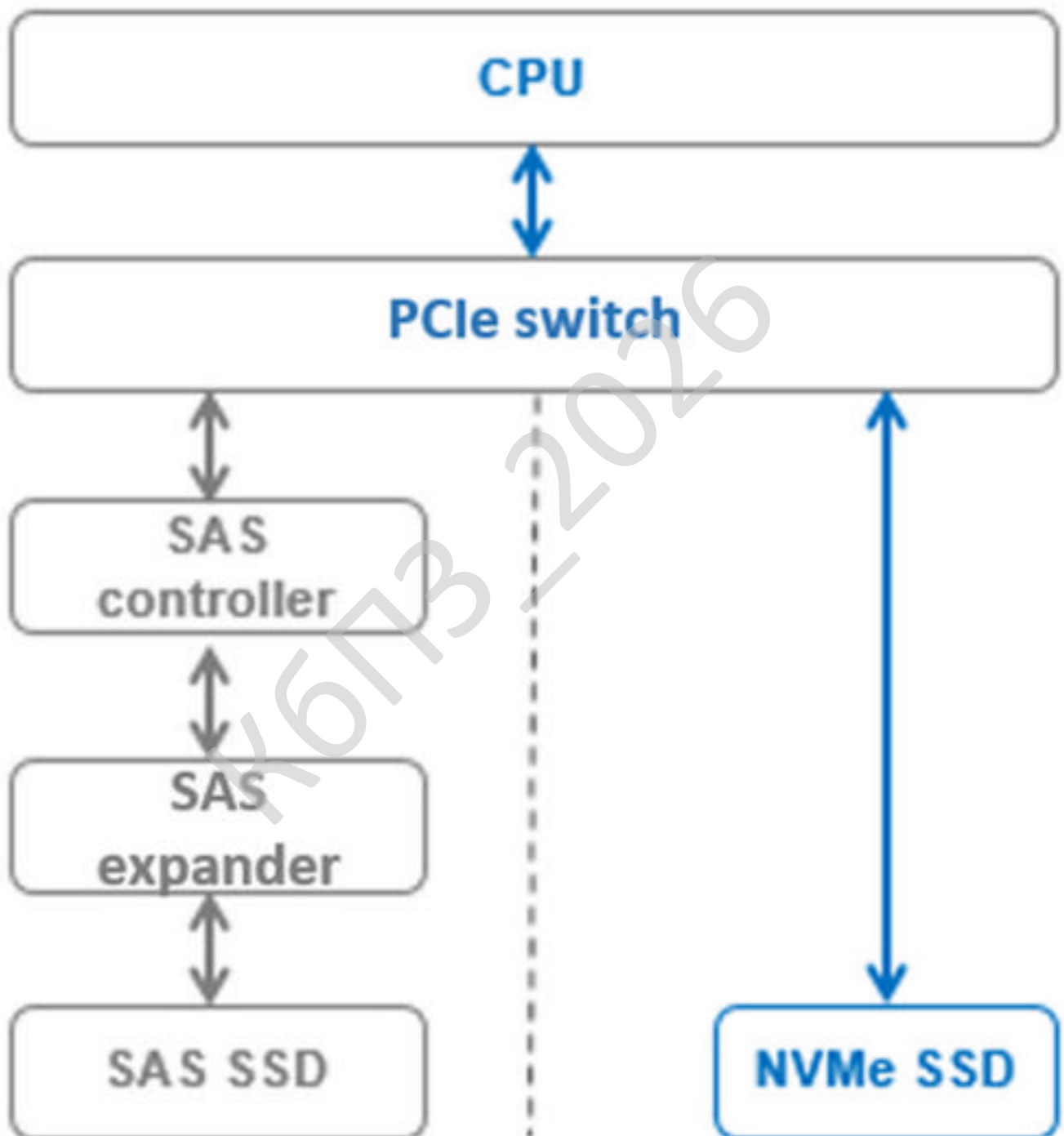


Рисунок 3.1 – Структурна схема системи

Флеш-масиви на базі NVMe передають дані вводу/виводу від процесора на твердотільний накопичувач через шину й комутатори PCIe. Процесори флеш-масивів на базі NVMe прямо взаємодіють із накопичувачами NVMe по найбільш короткому шляху, що підвищує ефективність і зменшує величину затримки при передачі даних.

По-друге, на аналітичному рівні програмного протоколу видні великі розходження між флеш-накопичувачами на базі SAS і NVMe в області технологій взаємодії протоколів для запису даних. Для завершення запиту на запис даних потрібно чотири взаємодії у випадку із протоколом SCSI (підключається через пристрій виводу SAS) і всього два для протоколу NVMe.

Розбір введення: SAS і NVMe

По-третє, на рівні інкапсуляції протоколу при використанні стека протоколів SAS запити вводу/виводу відправляються із блокових пристроїв на накопичувач по каналах SAS після інкапсуляції двох рівнів (протоколи SCSI і SAS). А при використанні стека протоколів NVMe для запитів введення-виводу потрібна інкапсуляція тільки одного рівня (протокол NVMe). Завдяки спрощеним стекам протоколів NVMe витрата ресурсів на інкапсуляцію знижується на 50%, тим самим зменшуючи навантаження на ЦП і затримку при передачі даних введення-виводу, викликувану кожною інкапсуляцією.

По-четверте, на рівні паралельної роботи декількох черг протокол SAS підтримує тільки одну чергу, тоді як протокол NVMe підтримує до 64 тис. черг, кожна з яких здатна обробляти до 64 тис. паралельних команд. Протокол NVMe з підтримкою множини черг працює краще завдяки обробці більшої кількості паралельних завдань і більше ефективній взаємодії декількох каналів і контролерів у твердотільному накопичувачі.

В-п'ятих, на рівні механізму блокування архітектуру SAS з однією чергою необхідно блокувати в багатоядерному середовищі. Розроблений механізм керування чергою операцій вводу/виводу для NVMe, що повністю виключає взаємовиключне блокування рівнів диска на вихідних маршрутах вводу/виводу й

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

усуває конфлікти обробки вводу/виводу. Цей механізм забезпечує повну сумісність із паралельною обробкою даних у багатоядерних процесорах, зменшує витрати на програмне забезпечення й підвищує ефективність внутрішньої обробки. При керуванні чергою операцій вводу/виводу багатопотокова обробка й безліч черг дозволяють досягти оптимальної продуктивності.

Керування чергою операцій вводу/виводу в системи NVMe

В-шостих, на рівні оптимізації ОС операційна система зберігання, спеціально розроблена для флеш-пам'яті, поліпшує алгоритм взаємодії дискових контролерів.

Стабільність і ефективність

Завдяки перерахованим перевагам клієнти, які захочуть вибрати новий протокол зберігання, швидше за все, звернуться до NVMe. Незважаючи на його безпрецедентну продуктивність, протокол NVMe припускає й нові труднощі при проектуванні дисків і систем зберігання.

– **Невласні порти з подвійним роз'ємом PCIe.** Технологія резервування із двома контролерами використовується в корпоративних сховищах для забезпечення надійності системи. Однак технологія із двома роз'ємами на базі PCIe викликає складності при проектуванні твердотільних накопичувачів. У цей час в SSD-накопичувачах деяких виробників не використовуються рідні подвійні порти PCIe. Наприклад, в NVMe SSD від Intel використовуються подвійні порти PCIe, підключаємі за допомогою внутрішніх комутаторів PCIe. Такі недоліки позначаються на надійності системи.

– **Низька продуктивність при гарячій заміні PCIe.** Комерційні SSD на базі NVMe повинні підтримувати функцію приєднання без відключення живлення. Однак традиційна гаряча заміна, особливо раптова, незручна для користувача через виникаючі системні виключення й переривання роботи пристрою.

– **Необхідність гарантувати надійність даних.** Оскільки на дисках може зберігатися усе більше даних, то для захисту таких обсягів потрібно більше висока надійність накопичувача.

– **Труднощі в забезпеченні надійності при розміщенні на різних об'єктах.** Питання надійності одиночних систем зберігання вже піднімався. Тепер він коштує й перед системами, розміщеними на різних об'єктах.

По-перше, на рівні інтерфейсу SSD-накопичувачі на базі NVMe використовують рідні подвійні порти із двома незалежними каналами PCIe 3.0. Це забезпечує апаратну основу для відновлення системи й обробки виключень, а також підтримку резервування із двома контролерами, що підвищує надійність корпоративних систем зберігання.

По-друге, є експертом в області телекомунікацій, особливо коли справа стосується комплексного керування каналами PCIe, технологій усунення несправностей PCIe і гарячої заміни. Драйвер PCIe розроблений таким чином, щоб твердотільний накопичувач можна було видалити в будь-який час. За рахунок цього забезпечується висока надійність системи PCIe при заміні диска або виникненні збоїти.

Механізм видалення SSD-накопичувача NVMe

По-третє, для захисту даних була використана інноваційна технологія RAID-TP, заснована на алгоритмі Erasure Code (EC). Біти парності підтримують 1, 2 і 3 виміру й можуть допускати вихід з ладу від одного до трьох дисків одночасно. Це означає, що навіть у випадку виходу з ладу трьох дисків система не постраждає від втрати даних або переривання обслуговування. На сьогоднішній день три одночасних збої можуть витримати тільки продукти Huawei, NetApp і Nimble, а інші постачальники (такі, як Dell EMC, HDS і IBM) поки не можуть цим похвастатися.

Хоча продукти деяких вендорів можуть працювати при збої трьох дисків, у них використовується традиційна архітектура RAID: фіксовані диски з даними й диски гарячого резерву. Відновлення 1 Тб даних за допомогою гарячого

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

резерву в пристроях цих компаній займає 5 годин. В OceanStor Dorado використовується система блокової віртуалізації, за допомогою якої дані в 1 ТБ можна відновити всього за 30 минут, що відповідає вимогам систем зберігання більших обсягів даних.

По-четверте, для захисту даних, розміщених на різних об'єктах у флеш-сховищах на базі NVMe використовуються комплексні технології захисту даних, такі як створення моментальних знімків, клонування й віддалена реплікація, які допоможуть клієнтам створити рішення для багатоступінчастого захисту даних як у локальному й міських, так і віддаленому центрах обробки даних (ЦОД).

Не відставайте від прогресу, підготуйтеся до майбутнього

Через великий час пошуку затримки в системах зберігання на базі традиційних жорстких дисків може перевищувати 10 мс. Завдяки застосуванню електронних таблиць зіставлення SSD-накопичувачі дозволяють зменшити час затримки у два рази до 5 мс.

Традиційні контролери запам'ятовувального пристрою часто використовують ті самі ОС, незважаючи на розходження у форматах жорстких дисків. Тому їх зручно використовувати при зміні типу диска. Однак безліч типів жорстких дисків і використовуваних на них ОС застаріли. Саме із цієї причини випустила флеш-масиви OceanStor Dorado V3. Ці системи розроблені спеціально для роботи з SSD і здатні зменшити затримки до значень нижче 1 мс.

У майбутньому багато компаній, що бажають витягти користь із нових способів зберігання даних, без сумніву перейдуть на більше швидкі носії. Цим і корисні сучасні технології.

Що таке NVMe SSD і чим вони краще звичайних SSD дисків?

Якщо ви задалися метою придбати SSD, то можете з подивом виявити, що орієнтуватися винятково на обсяг не доводиться, ціnnики за просто можуть відрізнятись у два, а те й три рази. Я вже небагато торкався даної теми, коли розповідав про характеристики надійності TBW і DWPD в SSD дисків. Але зв'язано це не тільки з надійністю.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

За останні кілька років, у технологіях виробництва SSD відбулася велика купа змін і з ходу вже не вийде розібратися у всій розмаїтості контролерів, інтерфейсів і форм-факторів сучасних SSD. Сьогодні спробу максимально просто розповісти, що таке NVMe SSD, які приходять на зміну звичайним SATA SSD і чи коштує, насправді, переплачувати за новітні технології.

Якщо зі звичайними жорсткими дисками все зрозуміло й інтерфейсу SATA 3.0 із пропускнуою здатністю в 600 Мбайт/с ним вистачає з головою, та й імовірність що вони колись доберуться до таких швидкостей практично нульова, то з SSD дисками зовсім інша історія. У рамках застаріваючого інтерфейсу SATA їм було тісно із самого початку й вони швидко досягли стелі.

Для розкриття всього потенціалу нових SSD дисків треба було перейти на пряме підключення до швидкісної шини PCI Express і зміні форм-фактора на M.2, що теоретично дозволяє досягати швидкостей до 3,94 Гбайт/с. За роботу накопичувача на шині PCI Express саме й відповідає NVMe-контролер, на відміну від SATA-дисків, які управляються AHCI-контролерами.

NVMe є скороченням від **NVM Express**. «NVM» у назві специфікації позначає енергонезалежну пам'ять (Non-Volatile Memory).

Вони можуть монтуватися безпосередньо на материнську плату або вставлятися у звичайний слот PCI Express через спеціальний переходник. Є навіть адаптери для установки накопичувачів M.2 на місце звичайних 2,5" дисків з інтерфейсом SATA. Один такий я використовую в роботі, коли доводиться мати справу з дисками формату M.2, підключаючи його як зовнішній через USB-Рознімання.

Багато NVMe-накопичувачів випускається у форм-факторі плат PCI Express, що особливо характерно для корпоративного сегмента.

Недорогі SSD накопичувачі у форм-факторі M.2 можна використовувати в якості високошвидкісний флешки або замінити ними звичайні диски за допомогою SATA адаптера, з відповідним швидкісним обмеженням в 600 Мбайт/с.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Неначебто стало зрозуміліше, але отут виникає інше питання, а чи варто додатково відкладати по рублі зі сніданків на придбання більше швидкісного NVMe SSD або можна обмежитися накопичувачем з інтерфейсом SATA, адже навіть він дає гарний приріст швидкості реакції комп'ютера в порівнянні зі звичайним HDD.

Якщо дивитися голі цифри, то швидкість сучасних NVMe SSD переступила за 2000 Мбайт/с (!) чи Значить це, що вони настільки ж ефективніше SSD з інтерфейсом SATA? І та й немає...

NVMe SSD у реальних додатках: чи варто неодмінно гнатися за новою технологією?

Відразу хочеться відзначити, що виконання накопичувача в сучасному компактному форм-факторі M.2 ще не гарантує що він побудований на контролері NVMe. Багато дисків початкового рівня, що випускаються у форм-факторі M.2, по колишньому управляються AHCI-контролерами. Перед покупкою, ознайомтеся з характеристиками і якщо побачите, що заявлена швидкість читання 550 Мбайт/с, а запису 330 Мбайт/с, те можна сміло затверджувати, що ніяким NVMe отут навіть не пахне.

Крім того, самі контролери NVMe, навіть одного виробника, можуть сильно розрізнятися й підтримувати тільки дві або повноцінні чотири лінії PCI Express. Все це теж впливає на ціну накопичувача. Ознакою високошвидкісного SSD диска формату M.2 може служити наявність швидкісного ключа M (виріз на платі) або відразу двох ключів – універсального B і M.

Втім, тестування в реальних робочих умовах, а не синтетичних тестах, начебто CrystalDiskMark, що замірить швидкісні характеристики в ідеальних умовах, уже не так вражають і накопичувачі поведуться набагато скромніше. Отут, звичайно, все залежить від того чим ви займаєтеся на комп'ютері.

Так, використання високошвидкісних NVMe SSD практично ніяк не впливає на швидкість рендерингу 3D-моделей, отут основне навантаження лягає на відеокарту й центральний процесор, а дані підкачуються з оперативки.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Зразкова така ж картина буде спостерігатися й в іграх, так що користі отут від NVMe буде мало.

У реальності, такі швидкості можуть знадобитися тільки при при обробці відео (Final Cut, iMovie, Adobe Premiere, After Effects) і важких іграх, але обов'язково в парі з потужним процесором щоб устигати прокачувати такий обсяг даних.

Який вивід можна зробити із усього сказаного? Якщо, при виборі накопичувача, ціна для вас не є визначальним аргументом або ви постійно працюєте з відео, то не роздумуючи беріть диски з NVMe. Як не крути, але накопичувачі з інтерфейсами SATA/SAS доживають своє століття й перехід на NVMe SSD неминучий через їх більше просту реалізацію й високу продуктивність.

В інших випадках переплачувати за NVMe SSD думаю не коштує, принципової різниці в роботі, у порівнянні з SSD дисками з інтерфейсом SATA, ви не заметете й вийде небагато заощадити.

3.3 Розробка функціональної схеми

Системи зберігання даних (СЗД)

Система зберігання даних (СЗД) являє собою комплекс програмних і апаратних засобів, створених для керування й зберігання більших обсягів інформації. Основними носіями інформації тепер є жорсткі диски, обсяги яких зовсім недавно досягли 1 терабайта. Основним сховищем інформації в малих компаніях є файлові сервери й сервери СУБД, дані яких зберігаються на локальних жорстких дисках. У великих компаніях обсяги інформації можуть досягати сотень терабайт, причому до них висуваються ще більші вимоги по швидкості й надійності. Ніякі локально підключені до серверів диски не можуть задовольнити цим потребам. Саме тому великі компанії впроваджують системи зберігання даних (СЗД).

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Основними компонентами СЗД є: носії інформації, системи керування даними й мережі передачі даних.

– **Носії інформації.** Як уже було сказано вище, зараз основними носіями інформації є жорсткі диски (можливо в найближчому майбутньому будуть замінені твердотільними електронними накопичувачами SSD). Жорсткі диски, підрозділяються на 2 основних типи: надійні й продуктивні SAS (Serial Attached SCSI) і більше економічні SATA. У системах резервного копіювання також застосовуються стрічкові накопичувачі (стріммери).

– **Системи керування даними.** СЗД надає потужні функції по керуванню даними. СЗД забезпечує функції дзеркалювання й реплікації даних між системами, підтримує відказостійкі, що самовідновлюються масиви, надає функції моніторингу, а також функції резервного копіювання на апаратному рівні.

– **Мережі передачі даних.** Мережі передачі даних надають середовище, по якій здійснюється зв'язок між серверами й СЗД або зв'язок однієї СЗД із іншої. Жорсткі диски розділяють по типі підключення: DAS (Direct Attached Storage) – безпосередньо підключені до сервера диски, NAS (Network Attached Storage) – диски, підключені по мережі (доступ до даних здійснюється на рівні файлів, звичайно по FTP, NFS або SMB) і SAN (Storage Area Network) – мережі зберігання даних (надають блоковий доступ). У великих системах зберігання даних основним типом підключення є SAN. Існує 2 методи побудови SAN на основі Fibre Channel і iSCSI. Fibre Channel (FC) в основному застосовується для з'єднання усередині одного центра обробки даних. А iSCSI являє собою протокол передачі SCSI команд поверх IP, які можуть маршрутизуватися звичайними IP маршрутизаторами. iSCSI дозволяє будувати гео-розподілені кластери.

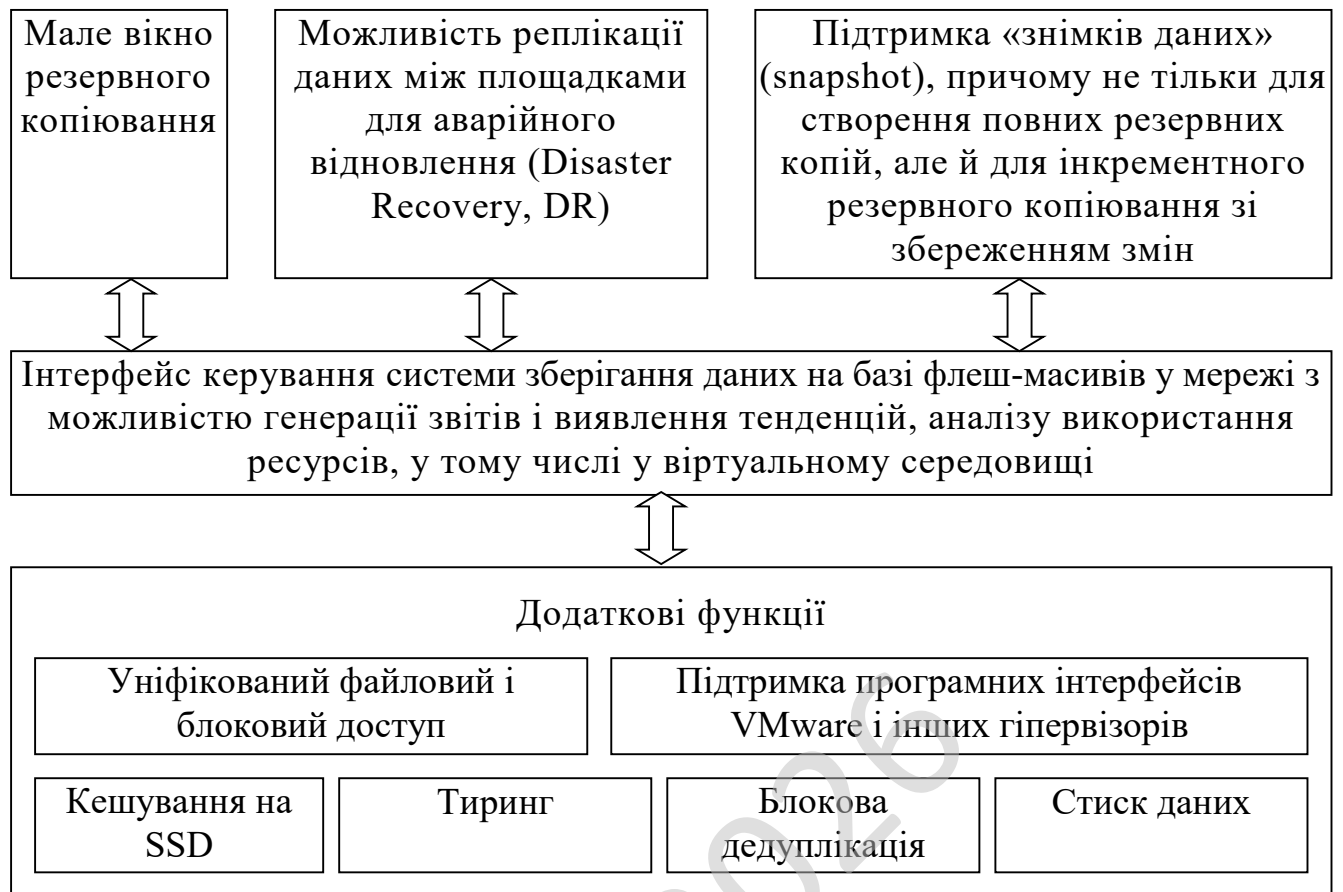


Рисунок 3.2 – Функціональна схема системи

У числі «обов'язкових» вимог, пропонованих до сучасних систем зберігання:

- Мале вікно резервного копіювання.
- Можливість реплікації даних між площадками для аварійного відновлення (Disaster Recovery, DR).
- Підтримка «знімків даних» (snapshot), причому не тільки для створення повних резервних копій, але й для інкрементного резервного копіювання зі збереженням змін.
- Простий інтерфейс керування з можливістю генерації звітів і виявлення тенденцій, аналізу використання ресурсів, у тому числі у віртуальному середовищі.

Додаткові функції:

- Уніфікований файловий і блоковий доступ: можливість підтримки файлового й блокового зберігання з єдиним інтерфейсом адміністрування (unified storage).
 - Кешування на SSD: флеш-пам'ять використовується для кешування даних масиву.
 - Тиринг: багаторівневе зберігання з автоматичною міграцією даних між рівнями. У якості одного з рівнів зберігання звичайно виступають твердотільні накопичувачі (SSD).
 - Підтримка програмних інтерфейсів VMware і інших гіпервізорів: спрощує інтеграцію масиву із середовищем віртуалізації.
 - Блокова дедуплікація: усуває надлишкові блоки даних і дозволяє значно скоротити потреби в ємності зберігання.
 - Стиск даних: дані кодуються таким чином, щоб займати меншу ємність. Іноді дані стискаються «на лету», перед збереженням на диск.
- Останні дві функції можуть сприяти істотному скороченню витрат.

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.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37



Рисунок 3.3 – Діаграма взаємодії процесів

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.

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. 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.

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

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

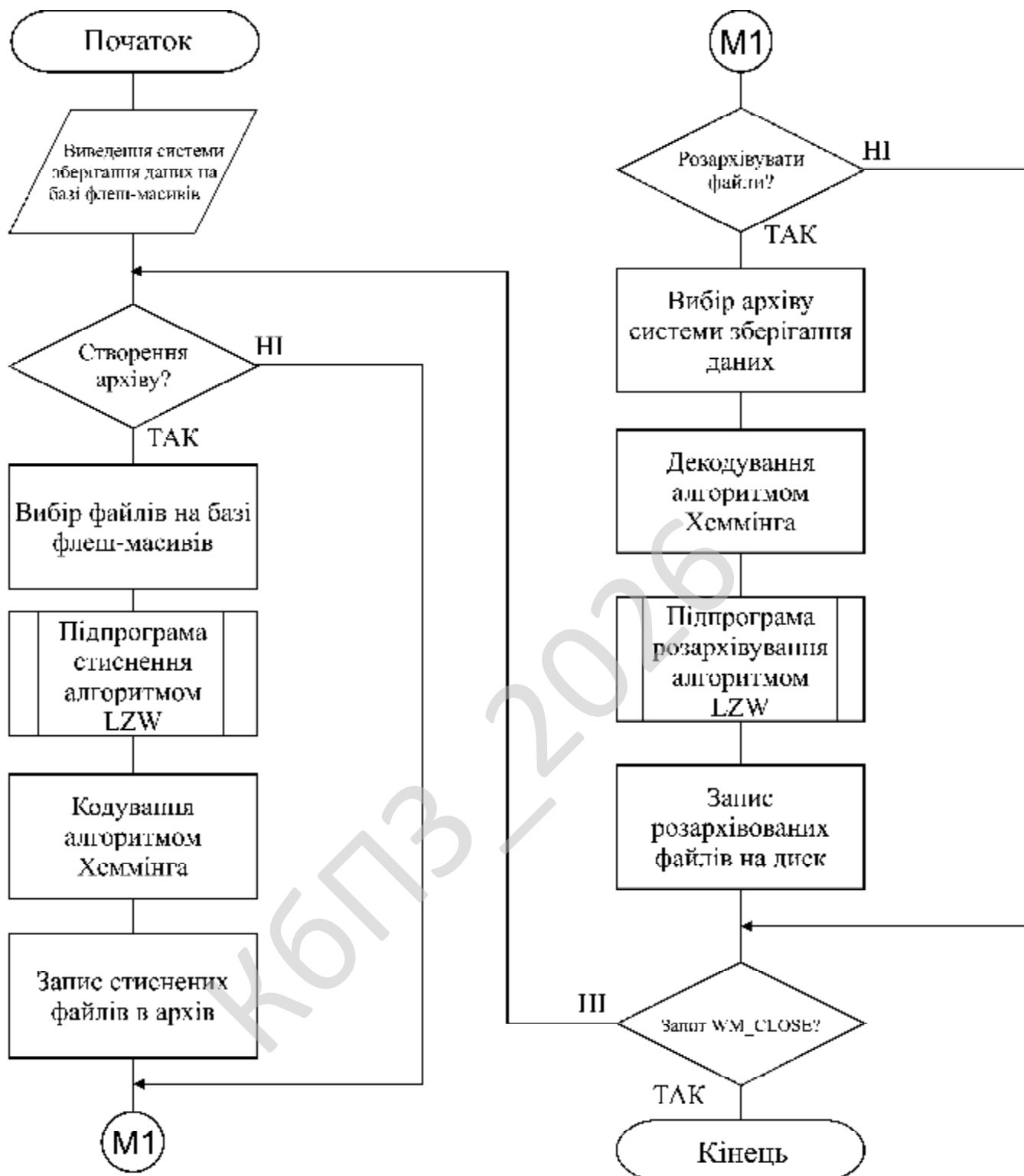


Рисунок 4.1 – Блок-схема основної програми

Було використано наступні підходи UML – діаграма діяльності (діаграми поведінки типу. Діаграма діяльності. Це візуальне представлення графу

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

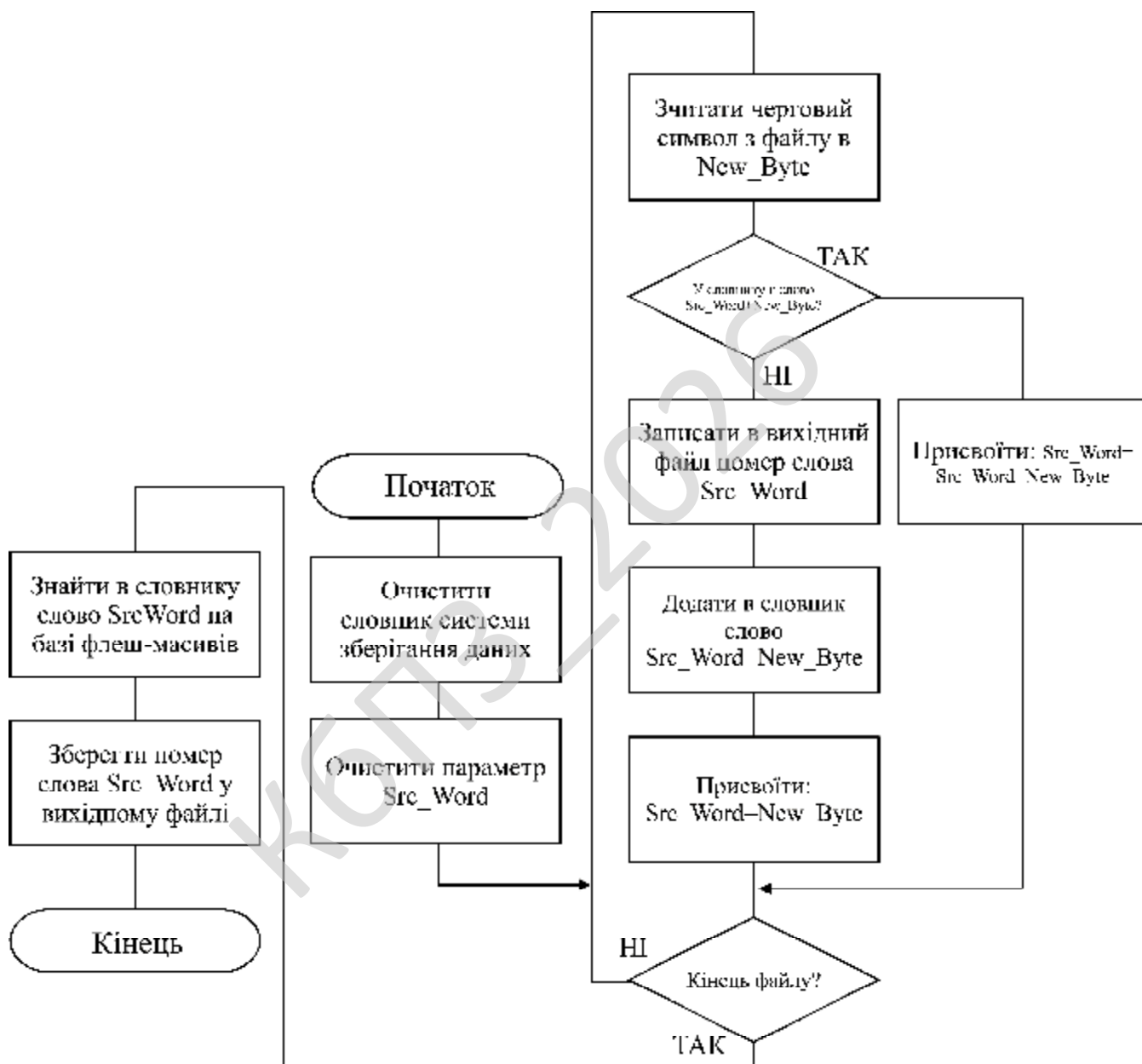


Рисунок 4.2 – Блок-схема роботи підпрограми

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання

діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

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

Система керування базами даних (СКБД, Database Management System, DBMS) – набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Першим поколінням СКБД прийнято вважати ієрархічні й мережеві системи. Ці системи отримали широке поширення в 1970-х роках, а першою комерційною системою цього типу була система IMS компанії IBM.

У 1980-х роках ці системи були витіснені системами другого покоління – повсюдно використовуваними і донині реляційними СКБД. У цих системах використовувалися непроцедурні мови управління даними (SQL) і передбачався значний ступінь незалежності даних. Реляційні системи внесли значні удосконалення в управління даними: графічний користувацький інтерфейс (GUI), клієнт-серверні застосунки, розподілені бази даних, паралельний пошук даних та інтелектуальний аналіз даних.

Але вже до кінця 1980-х років існуюча тоді реляційна модель перестала задовольняти розробників в силу низки обмежень. Відповіддю на зростаючу складність програм баз даних стали два нових напрямки розвитку СКБД: об'єктно-орієнтовані СКБД і об'єктно-реляційні СКБД.

У 1991 був утворений консорціум ODMG, основною метою якого стало вироблення промислового стандарту об'єктно-орієнтованих баз даних. Між 1993 та 2001 роками ODMG опублікувала п'ять ревізій своїх специфікацій. Остання

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

версія стандарту має індекс 3.0, після чого група розпустилася. До кінця 1990-х існувало близько десяти компаній, що виробляли комерційні продукти, що позиціонуються на ринку як ООСКБД. Найбільш відомими системами даного класу стали Objectivity, Versant виробництва однойменних компаній, а також СКБД Jasmine, випущена компанією CA. Незважаючи на переваги, що дозволяють ефективніше вирішувати певний ряд завдань, об'єктно-орієнтовані системи так і не змогли завоювати значущу частку ринку СКБД, залишившись «нішевим» продуктом.

Постачальниками традиційних реляційних СКБД також була проведена значна робота з об'єднання об'єктно-орієнтованих і реляційних систем. Розробники постаралися розширити мову SQL, щоб включити в неї концепції об'єктно-орієнтованого підходу, зберігаючи переваги реляційної моделі (об'єктні розширення мови SQL були зафіксовані в стандарті SQL:1999).

Основний принцип – це еволюційний розвиток можливостей СКБД без поломки попередніх підходів та зі збереженням наступності з системами попереднього покоління.

Поняття СКБД третього покоління, якими, власне кажучи, і є об'єктно-реляційні СКБД, з'явилося після опублікування групою відомих фахівців в області баз даних «Маніфесту систем баз даних третього покоління». Основні принципи СКБД третього покоління, позначені в маніфесті. Крім традиційних послуг з управління даними, СКБД третього покоління повинні забезпечити підтримку розвиненіших структур об'єктів і правил. Розвинутіша структура об'єктів характеризує засоби, необхідні для зберігання і маніпулювання нетрадиційними елементами даних (тексти, просторові дані, мультимедіа).

СКБД третього покоління повинні включити в себе СКБД другого покоління. Системи другого покоління внесли вирішальний вклад у двох областях – непроцедурний доступ за допомогою мови запитів SQL і незалежність даних. Ці досягнення обов'язково повинні враховуватися в системах третього покоління.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

СКБД третього покоління повинні бути відкриті для інших підсистем. Це включає оснащення різноманітними інструментами підтримки прийняття рішень, доступом з багатьох мов програмування, інтерфейсами до існуючих популярних систем і бізнес-застосунків, можливістю запуску програм з бази даних на іншій машині і розподілені СКБД. Весь набір інструментів і СКБД має ефективно функціонувати на різноманітних апаратних платформах з різними операційними системами.

Крім того, СКБД, що розраховує на широку сферу застосування, повинна бути оснащена мовою четвертого покоління (4GL).

У середині 1990 років було лише кілька дослідних прототипів СКБД, які поєднали найкращі риси реляційних і об'єктно-орієнтованих СКБД. Першим комерційним продуктом, якому були властиві об'єктно-реляційні риси, став Universal Server компанії Informix(згодом була поглинена IBM). В даний час більшість цих ідей вже втілено в реальних комерційних рішеннях, в тому числі і в продуктах основних постачальників СКБД (Oracle Database і IBM DB2).

Розвиток індустрії систем керування базами даних базується на значних фундаментальних наукових дослідженнях. Найчастіше, між самими дослідженнями та їхньою конкретною реалізацією в прикладних рішеннях минають роки, а іноді й десятиліття. Роботу в області управління даними проводять як університетські дослідницькі групи (MIT, Berkeley), так і центри розробок основних постачальників СКБД (Oracle, IBM, Microsoft). Інвестування в управління даними – це довгострокове, і разом з тим, вигідне вкладення коштів. В даний час дослідники мають у своєму розпорядженні засоби, що дозволяють ефективно реалізувати найскладніші запити, що маніпулюють терабайтами й петабайтами різних даних.

Основними тенденціями, які дали привід для проведення різних масштабних досліджень в області баз даних стали:

Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число

прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.

Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповнятися численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.

Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно ми спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової оперативної пам'яті.

Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв.

Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом – з паралельними обчисленнями і хмарними технологіями.

Основні характеристики СКБД

1. Контроль за надлишковістю даних.
2. Несуперечливість даних.
3. Підтримка цілісності бази даних (коректність та несуперечливість).
4. Цілісність описується за допомогою обмежень.
5. Незалежність прикладних програм від даних.
6. Спільне використання даних.
7. Підвищений рівень безпеки.

Можливості СКБД

1. Дозволяється створювати БД (здійснюється за допомогою мови визначення даних DDL (Data Definition Language)).

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

2. Дозволяється додавання, оновлення, видалення та читання інформації з БД (за допомогою мови маніпулювання даними DML, яку часто називають мовою запитів)

3. Можна надавати контрольований доступ до БД за допомогою: Системи забезпечення захисту, яка запобігає несанкціонованому доступу до БД; Системи керування паралельною роботою прикладних програм, яка контролює процеси спільного доступу до БД; Система відновлення – дозволяє відновлювати БД до попереднього несуперечливого стану, що був порушений в результаті збою апаратного або програмного забезпечення.

Основні компоненти середовища СКБД

1. Апаратне забезпечення.
2. Програмне забезпечення.
3. Дані.
4. Процедури – інструкції та правила, які повинні враховуватись при проектуванні та використанні БД.
5. Користувачі: адміністратори даних (керування даними, проектування БД, розробка алгоритмів, процедур) та БД (фізичне проектування, відповідальність за безпеку та цілісність даних); розробники БД; прикладні програмісти; кінцеві користувачі.

Архітектура СКБД

Існує трирівнева система організації СКБД ANSI-SPARC, при якій існує незалежний рівень для ізоляції програми від особливостей представлення даних на нижчому рівні.

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.
2. Концептуальний – узагальнене представлення БД, описує які дані зберігаються в БД і зв'язки між ними. Підтримує зовнішні представлення, підтримується внутрішнім рівнем.
3. Внутрішній – фізичне представлення БД в комп'ютері.

Логічна незалежність – повна захищеність зовнішніх моделей від змін, що вносяться в концептуальну модель.

Фізична незалежність – захищеність концептуальної моделі від змін, які вносяться у внутрішню модель.

SQLite – полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92.

Сирцевий код SQLite поширюється як суспільне надбання (public domain), тобто може використовуватися без обмежень та безоплатно з будь-якою метою. Фінансову підтримку розробників SQLite здійснює спеціально створений консорціум, до якого входять такі компанії, як Adobe, Oracle, Mozilla, Nokia, Bentley і Bloomberg.

З 2018р SQLite, як й JSON та CSV, рекомендований Бібліотекою Конгресу США формат зберігання структурованого набору даних. У 2005 році проект отримав нагороду Google-O'Reilly Open Source Awards.

Особливістю SQLite є те, що вона не використовує парадигму клієнт-сервер, тобто рушій SQLite не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми.

Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму.

SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується застосунок.

Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Кілька процесів або потоків можуть одночасно без жодних проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

жодних інших запитів у цей час не обслуговується; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу `sqlite3`, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка, і дозволяє звертатися до файлу БД на основі типових функцій ОС.

Завдяки архітектурі рушія можливо використовувати SQLite як на вбудовуваних (embedded) системах, так і на виділених машинах з гігабайтними масивами даних.

Особливості SQLite:

1. Транзакції атомарні, послідовні, ізольовані, і міцні (ACID) навіть після збоїв системи і збоїв живлення.
2. Встановлення без конфігурації – не потребує ані установки, ані адміністрування.
3. Реалізує значну частину стандарту SQL92.
4. База даних зберігається в одному крос-платформовому файлі на диску.
5. Підтримка терабайтних розмірів баз даних і гігабайтного розміру рядків і BLOBів.
6. Малий розмір коду: менше ніж 350KB повністю налаштований, і менш 200KB з опущеними додатковими функціями.
7. Швидший за популярні рушії клієнт-серверних баз даних для найпоширеніших операцій.
8. Простий, легкий у використанні API.
9. Написана в ANSI C, включена прив'язка до TCL; доступні також прив'язки для десятків інших мов.
10. Добре прокоментований сирцевий код зі 100% тестовий покриттям гілок.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

11. Доступний як єдиний файл сирцевого коду на ANSI C, який можна легко вставити в інший проект.

12. Автономність: немає зовнішніх залежностей.

13. Крос-платформовість: з коробки підтримується Unix (Linux і Mac OS X), OS/2, Windows (Win32 і WinCE). Легко переноситься на інші системи.

14. Сирці перебувають в суспільному надбанні.

15. Поставляється з автономним клієнтом інтерфейсу командного рядка, який може бути використаний для управління базами даних SQLite.

Інструменти створення та обслуговування БД

Створення та обслуговування БД можуть здійснюватись через текстову консоль SQL-командами або через спеціальні інструменти, у тому числі – з графічним інтерфейсом користувача.

Технології, що підтримують SQLite та мови програмування

Сама бібліотека SQLite написана мовою C. Проте є реалізація бібліотеки на JavaScript sql.js, яка дозволяє обробляти файли БД безпосередньо в браузері.

Для інших мов програмування розроблено механізм підключення й роботи з БД через цю бібліотеку: C++, Java, Python, Perl, PHP, Ruby, Haskell, Scheme, Smalltalk, Lua тощо. Засоби для роботи з Tcl включені в комплект постачання SQLite. Повний список наявних засобів можна знайти на сторінці проекту.

Web-інструментарії

У ряді інструментаріїв присутня можливість використання SQLite як бази даних, наприклад: Django; Java; PHP; Ruby on Rails; Trac; Symfony; Parser.

Багато програм підтримують SQLite як формат зберігання даних, зокрема:

1. Amarok – може використовувати бази даних SQLite як сховище музичної колекції.
2. Gajim – SQLite використовується для зберігання історії контактів.
3. Songbird (як застосунок, заснований на XULRunner 1.9)
4. Banshee.
5. F-Spot.

6. Платформа XUL на рушії Gecko 1.9, XULRunner 1.9 і, потенційно, всі застосунки, засновані на цій платформі, у тому числі й Firefox починаючи з версії 3.0.

7. Google Chrome.

8. Google Gears.

9. Mendeley – менеджер pdf-документів, академічний засіб для дослідження (реалізується desktop & web).

10. Zotero – менеджер інформації, бібліографічний менеджер, додаток Firefox.

Peer-to-peer (рівний до рівного) – варіант архітектури системи, в основі якої стоїть мережа рівноправних вузлів.

Комп'ютерні мережі типу peer-to-peer (або P2P) засновані на принципі рівноправності учасників і характеризуються тим, що їх елементи можуть зв'язуватися між собою, на відміну від традиційної архітектури, коли лише окрема категорія учасників, яка називається серверами може надавати певні сервіси іншим.

Фраза «peer-to-peer» була вперше використана у 1984 році Парбауелом Йохнухуйтсманом (Parbawell Yohnuhuitsman) при розробці архітектури Advanced Peer to Peer Networking фірми IBM.

В чистій «peer-to-peer» мережі не існує поняття клієнтів або серверів, лише рівні вузли, які одночасно функціонують як клієнти та сервери по відношенню до інших вузлів мережі. Ця модель мережевої взаємодії відрізняється від клієнт-серверної архітектури, в якій зв'язок відбувається лише між клієнтами та центральним сервером.

Така організація дозволяє зберігати працездатність мережі при будь-якій конфігурації доступних її учасників. Проте практикується використання P2P мереж які все ж таки мають сервери, але їх роль полягає вже не у наданні сервісів, а у підтримці інформації з приводу сервісів, що надаються клієнтами мережі.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

В P2P системі автономні вузли взаємодіють з іншими автономними вузлами. Вузли є автономними в тому сенсі, що не існує загальної влади, яка може контролювати їх. В результаті автономії вузлів, вони не можуть довіряти один одному та покладатися на поведінку інших вузлів, тому проблеми масштабування та надмірності стають важливішими ніж у випадку традиційної архітектури.

Сучасні P2P-мережі набули розвитку завдяки ідеям, пов'язаними з обміном інформацією, які формувалися у руслі того, кожен вузол може надавати та отримувати ресурси які надаються будь-якими іншими учасниками. У випадку мережі Napster, це був обмін музикою, в інших випадках це може бути надання процесорного часу для пошуку інопланетних цивілізацій (SETI@home) або ліків від раку (Folding@home).

Переваги P2P

Розподіл/зменшення вартості. Централізовані системи, які обслуговують багато клієнтів, зазвичай складають більшість вартості системи. Коли, ця вартість стає дуже великою, архітектура P2P може допомогти розподілити вартість серед користувачів. Наприклад, серед систем файлообміну Napster дозволив розподілити вартість зберігання файлів і міг підтримувати індекс, потрібний для сумісного використання. Економія коштів, здійснюється за допомогою використання та об'єднання ресурсів, які в іншому випадку не використовуються (наприклад SETI@home). Оскільки вузли зазвичай є автономними, важливо розподіляти витрати справедливо.

Об'єднання ресурсів. Децентралізований підхід веде до об'єднання ресурсів. Кожен вузол в системі P2P приносить певні ресурси як наприклад обчислювальна потужність або пам'ять. У програмах, які потребують величезну кількість цих ресурсів, як наприклад intensive моделювання або розподілені файлові системи, природно використовувати P2P, щоб залучити ці ресурси. Розподілені обчислювальні системи, як наприклад SETI@Home, distributed.net, і Endeavours – очевидні приклади цього підходу. Об'єднуючи ресурси тисяч вузлів,

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

вони можуть виконувати важкі з точки зору кількості обчислень функції. Файлобмінні системи, як наприклад Napster, Gnutella, і так далі, також об'єднують ресурси. У цих випадках, це дисковий простір, щоб зберігати дані, та пропускна спроможність, щоб їх передавати.

Вдосконалена масштабованість/надійність. З відсутністю сильної центральної влади по відношенню до автономних вузлів, важливою метою є покращення масштабованості і надійності. Масштабованість і надійність визначаються в традиційному для розподілених систем сенсі, як наприклад використання пропускної спроможності – скільки вузлів можуть бути досягнуті від одного вузла, скільки вузлів може підтримуватися, скільки користувачів може підтримуватися. Розподілена природа peer-to-peer мереж також збільшує помилкостійкість у разі невдач, шляхом дублювання даних поміж багатьох вузлів, і – в чистих системах P2P – надаючи можливість вузлу знайти дані без залежності від єдиного централізованого індексного сервера. У останньому випадку, немає ніякої єдиної критичної точки в системі.

Збільшена автономія. У багатьох випадках, користувачі розподіленої системи не бажають залежати від будь-якого централізованого постачальника послуг. Натомість, вони воліють, щоб всі дані та призначена для них робота виконувалась локально. Системи P2P підтримують цей рівень автономії, тому що вони вимагають, щоб кожен вузол робив необхідну для нього частину праці.

Анонімність/конфіденційність. Пов'язаним із автономією є поняття анонімності і конфіденційності. Користувач, можливо, не хоче, щоб кого-небудь або будь-який постачальник послуг знав про нього або про його роль у системі. З центральним сервером, гарантувати анонімність важко, тому що сервер зазвичай зможе ідентифікувати клієнта, як мінімум через його адресу в Інтернет. Використовуючи структуру P2P, в якій дії виконуються локально, користувачі можуть уникати необхідності передавати будь-яку інформацію про себе до кого-небудь іншого. FreeNet – яскравий приклад того, як анонімність може вбудуватися в додаток P2P. Він пересилає повідомлення через інші вузли, щоб

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

забезпечити неможливість вистежування початкового автора. Це збільшує анонімність, використовуючи ймовірнісні алгоритми таким чином, щоб походження не можливо було легко відстежити аналізуючи трафік у мережі.

Динамічність. Системи P2P припускають, що оточення надзвичайно динамічне. Тобто, ресурси, як наприклад вузли, з'являються та зникають із системи безперервно. У випадках комунікації, як наприклад мережі для обміну повідомленнями, використовуються так званий «список контактів», щоб інформувати користувачів, коли їхні друзі стають доступними. Без цього, потрібно було би, щоб користувачі «опитували» партнерів, посилаючи періодичні повідомлення. У випадку розподілених обчислень, як наприклад distributed.net і SETI@home, система повинна пристосуватись до заміни учасників. Тому вони повинні повторно видавати завдання для обчислення іншим учасникам, щоб гарантувати, що робота не втрачена, якщо попередні учасники відпадають від мережі, поки вони виконували крок обчислення.

Класифікація P2P систем

За функціями:

1. Розподілені обчислення. Обчислювальна проблема розподіляються на невеликі незалежні частини. Обробка кожної з частин робиться на індивідуальному ПК і результати збираються на центральному сервері. Цей центральний сервер відповідальний за розподілення елементів роботи серед окремих комп'ютерів в Інтернеті. Кожен із зареєстрованих користувачів має клієнтське програмне забезпечення. Воно користується періодами бездіяльності в ПК (часто це характеризується часами активації скрінсейверів), щоб виконувати деяке обчислення, надане сервером. Після того, як обчислення закінчене, результат посилається назад до сервера, і нова робота передається для клієнта.

2. Файлообмін. Зберігання та обмін даними - це одна з областей, де технологія P2P була найуспішнішою. Мультимедійні дані, наприклад, вимагають великих файлів. Napster і Gnutella використовувались користувачами, щоб обійти

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

обмеження пропускної спроможності, які роблять передачу великих файлів неприйнятними.

3. Співпраця. Природа технології P2P робить її добре придатною для забезпечення співпраці між користувачами. Це може бути обмін повідомленнями, онлайн ігри, сумісна робота над документами в бізнесі, освіті та дома.

За ступенем централізації:

1. Чисті peer-to-peer системи. Вузли є рівними, поєднуючи ролі серверу та клієнту. Не існує центрального сервера, що керує мережею. Прикладами таких систем є Gnutella та Freenet

2. Гібридні peer-to-peer системи. Мають центральний сервер, що зберігає інформацію про вузли та відповідає на запити відносно цієї інформації. Вузли займаються забезпеченням ресурсами (тому що центральний сервер їх не має), повідомленням сервера про наявність цих ресурсів надання ресурсів іншим вузлам, які бажають ними скористатися.

В залежності від того, як вузли з'єднуються один з одним можна поділити мережі на структуровані та неструктуровані:

1. Неструктурована мережа P2P формується, коли з'єднання встановлюються довільно. Такі мережі можуть бути легко сконструйовані, оскільки новий вузол, який хоче приєднатися до мережі, може скопіювати існуючі з'єднання іншого вузла, а вже потім почати формувати свої власні. У неструктурованій мережі P2P, якщо вузол бажає знайти певні дані в мережі, запит доведеться передати майже через всю мережу, щоб охопити так багато вузлів, як можливо. Головним недоліком таких мереж є те, що запити, можливо, не завжди вирішуються. Скоріш за все популярні дані будуть доступні в багатьох вузлів та пошук швидко знайде потрібне, але якщо вузол шукає рідкісні дані, наявні лише в декількох інших вузлів, то надзвичайно малоймовірно, що пошук буде успішним. Оскільки немає ніякої кореляції між вузлами та даними, що вони зберігають, немає ніякої гарантії, що запит знайде вузол, який має бажані дані.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

2. Структурована мережа P2P використовує єдиний алгоритм, щоб гарантувати, що будь-який вузол може ефективно передати запит іншому вузлу, який має бажаний файл, навіть якщо файл надзвичайно рідкісний. Така гарантія потребує структуровану систему з'єднань. У наш час найпопулярнішим типом структурованої мережі P2P є розподілені хеш-таблиці, в яких хешування використовується для встановлення зв'язку між даними та конкретним вузлом, який за них відповідає.

Було використано метод розробки динамічних систем (Dynamic Systems Development Method, DSDM) – це головним чином методика розробки програмного забезпечення, що базується на концепції швидкої розробки додатків (Rapid Application Development, RAD). У 2007 році **DSDM** став основним підходом до управління проектом і розробки додатків. DSDM – це ітеративний і інкрементний підхід, який надає особливого значення тривалого участі в процесі користувача/споживача.

Мета методу – здати готовий проект вчасно і вкластися в бюджет, але в той же час регулюючи зміни вимог до проекту під час його розробки. DSDM входить в сімейство гнучкої методології розробки програмного забезпечення, а також розробок, що не входять у сферу інформаційних технологій.

Остання версія DSDM називається DSDM Atern. Назва Atern – це скорочення від Arctic Tern (пер. Полярна крачка). Полярна крачка – птиця, яка може подорожувати на великі відстані. Вона символізує безліч аспектів методу, наприклад визначення пріоритету або кооперування, які є природним способом ведення робочого процесу.

Попередня версія DSDM (випущена в травні 2003 року), яка все ще діє і широко використовується, – це DSDM 4.2, є дещо розширеною версією DSDM 4. Розширена версія містить настанови щодо того, як використовувати DSDM спільно з Екстремальним програмуванням (Extreme Programming).

Як розширення концепції швидкої розробки додатків, DSDM фокусується на проектах інформаційних систем, що характеризуються стислими термінами і

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

бюджетами. У DSDM присутні вказівки на типові помилки проектів інформаційних систем, таких як перевищення бюджету, запізнення з термінами здачі (виконання), недостатнє залучення користувачів і топ-менеджерів в роботу над проектом. DSDM складається з:

- трьох стадій: передпроектна стадія життєвого циклу проекту і постпроектна стадія;
- стадія життя проекту складається з 5 етапів: дослідження реалізованості, дослідження економічної доцільності, створення функціональної моделі, проектування і розробка, етап реалізації.

При деяких умовах існує можливість включення в DSDM частин інших методик, таких як Rational Unified Process (RUP), Екстремальне програмування, PRINCE2. Інший гнучкий метод, схожий на DSDM по процесу і концепції – Scrum.

Метод DSDM був розроблений у Великобританії в 1990-х Консорціумом DSDM. Консорціум DSDM – це асоціація розробників та експертів в області програмного забезпечення, створена з метою «спільної розробки і просування незалежного фреймворку RAD» комбінуванням кращого практичного досвіду учасників асоціації. Консорціум DSDM – це некомерційна організація незалежних розробників, які володіють та управляють фреймворком DSDM. Перша версія фреймворку була завершена в січні 1995 року і опублікована в лютому 1995 року. У липні 2006 року була представлена відкрита версія DSDM 4.2, яка стала доступна приватним особам для перегляду і використання. Тим не менш, всі, хто поширює DSDM, повинні бути членами цього некомерційного консорціуму.

На початку 1990-х в індустрії інформаційних технологій став поширюватись новий термін – швидка розробка додатків (Rapid Application Development, RAD). Інтерфейси прикладних програм еволюціонували від старих зелених екранів до графічних інтерфейсів користувача, які використовуються і зараз. На ринок почали виходити нові інструменти для створення додатків, наприклад PowerBuilder. Вони дозволили розробникам простіше ділитися

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

планованими розробками з покупцями – з'явилося прототипування і почалося руйнування класичних, послідовних (каскадних) методів розробки.

Тим не менш, новий рух RAD було дуже неструктурованим: не існувало узгодженого опису цього методу і у багатьох організацій були створені власні опису і підходи до нього. Безліч великих корпорацій були зацікавлені в перспективах, що надаються методом, але вони також були стурбовані тим, щоб не знизився рівень якості їх продукції в кінцевому результаті.

Консорціум DSDM був утворений в 1994 році, коли група людей зустрілася на заході, організованому Butler Group в Лондоні. Всі, хто прийшов на цей захід, працювали у великих організаціях, таких як British Airways, American Express, Oracle and Logica (такі компанії як Data Sciences і Allied Domecq з тих пір були поглинені іншими організаціями).

На цьому зібранні було вирішено, що Дженніфер Степлтон, тоді представляла компанію Logica, розробить архітектуру комплексного, орієнтованого на користувача методу з хорошим контролем якості для ітеративної і інкрементної розробки. Підсумкова архітектура була спроектована так, щоб бути повністю сумісна зі стандартом ISO 9000 і PRINCE2. Коли архітектура була готова (через місяць після зборів), Консорціум сформував групи для її поширення у всіх областях розробки програмного забезпечення, які включали в себе: методи та засоби управління проектом, контроль якості та тестування, методи і засоби розробки. Контрольна група, очолювана творцем архітектури і складається з глав цих груп, повинна була забезпечити розуміння методу так, як він спочатку замислювався.

Незважаючи на те, що багато членів Консорціуму були прямими конкурентами, вони вільно ділилися тим, як вони вирішують проблеми, що виникають. Практика показала, що найкращий результат може бути досягнутий тільки працюючи як одне ціле. Консорціум збільшився за перший рік від декількох організацій до шістдесяти; опис методу ставало все більш і більш повним. Версія 1 була сформована в грудні 1994 року і опублікована в лютому

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

1995 року. Результатом став універсальний метод, що охоплює людей, процеси та інструменти. Він сформувався на основі досвіду організацій, різних за родом своєї діяльності і розмірами.

Метод DSDM – принципи. Існує 9 принципів, що складаються з 4 основних та 5 початкових точок.

1. Залучення користувача – це основа ведення ефективного проекту, де розробники ділять з користувачами робочий простір і тому прийняті рішення будуть більш точними.

2. Команда повинна бути уповноважена приймати важливі для проекту рішення без узгодження з начальством.

3. Часта поставка версій результату, з урахуванням такого правила, що «поставити щось хороше раніше – це завжди краще, ніж поставити все ідеально зроблена в кінці». Аналіз поставок версій з попередньої ітерації враховується на наступній.

4. Головний критерій – як можна більш швидка поставка програмного забезпечення, яке відповідає поточним потребам ринку. Але в той же час постачання продукту, який задовольняє потребам ринку, не менш важлива, ніж вирішення критичних проблем у функціоналі продукту.

5. Розробка – ітеративна та інкрементна. Вона ґрунтується на зворотного зв'язку з користувачем, щоб досягти оптимальної з економічної точки зору рішення.

6. Будь-які зміни під час розробки – оборотні.

7. Вимоги встановлюються на високому рівні перш, ніж почнеться проект.

8. Тестування інтегровано в життєвий цикл розробки.

9. Взаємодія і співпраця між усіма учасниками необхідно для його ефективності.

Передумови для використання DSDM.

Щоб успішно використовувати DSDM, необхідно щоб був виконаний ряд передумов. По-перше, необхідно організувати взаємодію між проектною

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

командою, майбутніми користувачами і вищим керівництвом. По-друге, повинна бути можливість поділу проекту на менші частини, що дозволить використовувати ітеративний підхід.

Два приклади проектів, для яких DSDM не дуже підходить:

1. Проекти, критичні безпеки розширене тестування та затвердження в таких проектах конфліктують з метою методу DSDM укластися в терміни і бюджет.

2. Проекти, чия мета зробити компоненти багаторазового використання – вимоги в таких проектах занадто високі і не вкладаються в принцип 80 %/20%.

Життєвий цикл проекту. Фреймворк DSDM складається з трьох послідовних стадій, які називаються передпроектна стадія, стадія життєвого циклу проекту і постпроектна стадія. Стадія життєвого циклу проекту – сама продумана і детально розроблена з усіх інших. Вона складається з п'яти етапів, які формують ітеративний, інкрементний підхід до розробки інформаційних систем.

Ці три фази і відповідні етапи будуть більш детально описані в наступних розділах. Для кожної стадії або етапи будуть розглянуті найважливіші функції і будуть представлені результати.

Стадія 1 – Передпроектна.

На цій стадії визначаються ймовірні проекти, відбувається виділення коштів та визначення проектної команди. Рішення задач на цій стадії допоможе уникнути проблем на більш пізніх стадіях проекту.

Стадія 2 – Життєвий цикл проекту.

Показано 5 етапів, які потрібно пройти проекту, щоб стати інформаційною системою. Перші два етапи, дослідження реалізованості та дослідження економічної доцільності, йдуть послідовно і доповнюють один одного. Після завершення цих етапів, відбувається ітеративна та інкрементна розробка системи в етапах: створення функціональної моделі, проектування і розробка, етап реалізації. Ітеративна та інкрементна природа DSDM буде описана далі.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Стадія 3 – Постпроектна.

На цій стадії забезпечується ефективна робота системи. Це досягається за рахунок підтримки проекту, його покращення та виправлення помилок згідно з принципами DSDM. Підтримка проекту здійснюється продовженням розробки, заснованої на ітеративної і інкрементній природі DSDM. Замість того, щоб закінчити проект за один цикл, зазвичай повертаються до попередніх стадій або етапів, щоб поліпшити продукт.

Нижче на діаграмі представлений весь життєвий цикл проекту. Ця діаграма описує ітеративну розробку DSDM. Опис кожного етапу буде представлено нижче.

Чотири етапи стадії життєвого циклу проекту.

Дослідження:

1. Дослідження реалізованості. На даному етапі визначається – потрапляє проект під рамки DSDM. Розглядаючи тип проекту, організаційні і кадрові питання, виносяться рішення – використовувати метод DSDM чи ні. Таким чином буде отримано звіт про застосовність, допустимий прототип і приблизний глобальний план проекту, який включає в себе план розробки і протокол можливих ризиків.

2. Дослідження економічної доцільності. На даному етапі аналізуються основні економічні і технологічні характеристики. Відбувається нарада експертів, на якій обговорюються найбільш важливі сторони системи і приймається рішення про пріоритети у розробці. На цьому етапі розробляються список основних вимог, опис сфери комерційної діяльності, опис архітектури системи і приблизний план створення прототипів.

Створення функціональної моделі:

1. Визначення функціонального прототипу. Визначення функціоналу, який буде закладено в прототипі на даному етапі. На цьому підетапі розробляється функціональна модель згідно до результатів, отриманих на стадії дослідження економічної доцільності.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

2. Узгодження планів. Відбувається узгодження як і в які терміни має бути розроблена функціональність прототипу.

3. Створення функціонального прототипу. Розробка функціонального прототипу, згідно з планами та функціональної моделі.

4. Аналіз функціонального прототипу. Перевірка справності розробленого прототипу. Це може бути досягнуто тестуванням прототипу кінцевим користувачем і/або переглядом документації. Результат – документ, що містить огляд функціонального прототипу.

Проектування і розробка:

1. Визначення конструктивного прототипу. Визначення функціональних і нефункціональних вимог системи. На основі цих вимог повинна бути створена стратегія реалізації. Якщо існує запис про тестування з попередньої ітерації, тоді він теж буде використаний для створення стратегії реалізації.

2. Узгодження планів. Відбувається узгодження як і в які терміни повинні бути реалізовані поставлені вимоги.

3. Створення конструктивного прототипу. Створення системи (конструктивного прототипу), яку можна віддати користувачам для тестування.

4. Аналіз конструктивного прототипу. Перевірити справність спроектованої системи. На цьому підетапі застосовується тестування та перегляд результатів. Створення користувацької документації та протоколу випробування.

Реалізація:

1. Затвердження системи користувачем. Кінцеві користувачі стверджують протестовану систему для подальшої реалізації і створення довідника користувача.

2. Навчання користувачів. Навчання майбутнього користувача роботі з системою. Результат підетапу – контингент підготовлених користувачів.

3. Реалізація. Реалізація протестованої системи серед користувачів.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

4. Аналіз ринку системи. Аналіз впливу випущеної системи на ринок. Головне питання – чи досягнута мета, поставлена при проектуванні системи. Грунтуючись на цьому проект переходить на наступну стадію (постпроектную) або повертається на попередню для доопрацювання. Аналіз буде представлений в документі аналізу проекту.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ММВ, в основі якого лежить змішування операцій різних алгебраїчних груп. ММВ – ітеративний алгоритм, що складається з лінійних дій (XOR і використання ключа) і паралельного застосування чотирьох великих оборотних нелінійних підстановок. Ці підстановки визначаються за допомогою множення по модулю $2^{32}-1$ з постійними множниками. У підсумку з'являється алгоритм, що використовує 128-бітовий ключ і 128-бітовий блок.

Алгоритм ММВ оперує 32-бітовими підблоками тексту (x_0, x_1, x_2, x_3) і 32-бітовими підблоками ключу (k_0, k_1, k_2, k_3) . Це спрощує реалізацію алгоритму на сучасних 64-бітових процесорах. Чергуючись із операцією XOR, шість разів використовується нелінійна функція f . Запишемо операції алгоритму (всі операції з індексами виконуються по модулю 4):

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_i \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0..3$$

$$f(x_0, x_1, x_2, x_3)$$

Функція f виконується в три кроки:

1. $x_i = c_i * x_i$ для $i = 0..3$ (Якщо на вході множення одні одиниці, то на виході – теж одні одиниці).

2. Якщо молодший значущий біт $x_0 = 1$, то $x_0 = x_0 \oplus C$. Якщо молодший значущий байт $x_3 = 0$, то $x_3 = x_3 \oplus C$.

3. $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$ для $i = 0..3$.

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі $2^{32}-1$. Спеціальний випадок для даного алгоритму: якщо другий операнд дорівнює $2^{32}-1$, результат теж дорівнює $2^{32}-1$. В алгоритмі використовуються наступні константи:

$$C = 2\text{aaaaaaa}, c_0 = 025\text{f1cdb}, c_1 = 2 * c_0, c_2 = 2^3 * c_0, c_3 = 2^7 * c_0.$$

Константа C – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом. У константи c_0 є інші особливі характеристики. Константи c_1 , c_2 і c_3 – зрушені версії c_0 , і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість c_i використовується c_i^{-1} . Значення $c_0^{-1} = 0\text{dad4694}$.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблена у бакалаврської дипломної роботі система зберігання даних на базі флеш-масивів у мережі. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

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

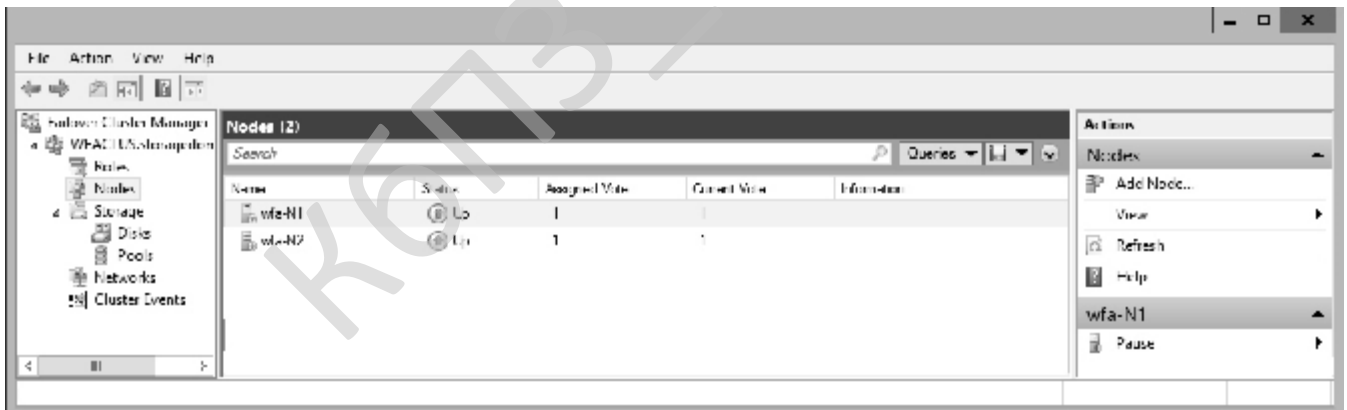


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких

помилки, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

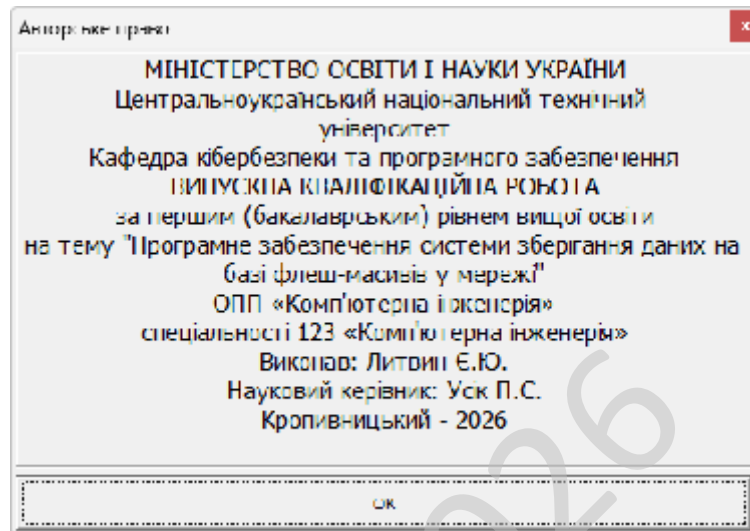


Рисунок 5.2 – Авторське право

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

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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 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.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

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», чию 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.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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

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

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

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

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

продукт.

Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення.

Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

КБПЗ_2026

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи зберігання даних на базі флеш-масивів у мережі.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем зберігання даних на базі флеш-масивів у мережі.
- Досліджена система зберігання даних на базі флеш-масивів у мережі.
- На основі отриманих результатів досліджень створена програмна реалізація системи зберігання даних на базі флеш-масивів у мережі.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання зберігання даних на базі флеш-масивів у мережі.

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 зберігання даних на базі флеш-масивів у мережі. 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

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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 ММВ.

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.

КБПЗ-2026

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

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-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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 користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.
16. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.
17. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

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.

					ВКРБ-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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-123.26.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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.