

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
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
**“Дослідження та програмна реалізація системи інтелектуальної
оптимізації продуктивності та надійності хмарних мережевих
систем зберігання даних”**

Виконав здобувач вищої освіти
II курсу, групи КН-24М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Кравченко В.О.
« ____ » _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Коваленко А.С.
« ____ » _____ 2025 р.

Рецензент _____

АНОТАЦІЯ

Кравченко В.О. Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Об'єктом дослідження є процес інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Предметом дослідження є методи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Методи дослідження базуються на методах теорії великих даних, теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

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 Visual C#.

Ключові слова: Комп'ютерні науки, оптимізація, продуктивність, надійність, хмарні мережеві системи зберігання даних

ABSTRACT

Kravchenko V.O. Research and software implementation of the system for intelligent optimization of the performance and reliability of cloud network data storage systems. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the system for intelligent optimization of the performance and reliability of cloud network data storage systems.

The purpose of the development is the research and software implementation of the system for intelligent optimization of the performance and reliability of cloud network data storage systems.

The object of the research is the process of intellectual optimization of the performance and reliability of cloud network data storage systems.

The subject of the research is the methods of intellectual optimization of the performance and reliability of cloud network data storage systems.

The research methods are based on the methods of big data theory, the theory of computer network construction, methods of mathematical statistics, and methods of software development.

The result of the work is a software implementation of a system for intelligent optimization of the performance and reliability of cloud network data storage systems.

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 the software are provided.

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

The program was developed in the Visual C# environment.

Keywords: computer science, optimization, performance, reliability, cloud network data storage systems

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	25
3.1 Опис функціонування системи	25
3.2 Розробка структурної схеми.....	29
3.3 Розробка функціональної схеми	33
3.4 Розробка діаграми процесів.....	45
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	47
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	47
4.2 Захист розробленого програмного забезпечення.....	63
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	67
6 НАУКОВА НОВИЗНА	73

						БКРМ-122.25.0043.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Кравченко В.О.</i>					М	1	98
<i>Перев.</i>	<i>Коваленко А.С.</i>					ЦНТУ КН-24М		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	74
7.1	Визначення цільової аудиторії кінцевого готового продукту	74
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	75
7.3	Вибір методу оцінки вартості ПЗ	75
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	76
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	78
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	79
7.7	Визначення ключових факторів успіху конкретного проєкту.....	79
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	81
8.1	Вступ.....	81
8.2	Аналіз умов праці	82
8.3	Техніка безпеки та протипожежна профілактика	86
8.4	Розробка заходів з охорони праці	88
8.5	Висновки до розділу.....	89
9	ОСНОВНІ ВИСНОВКИ.....	90
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	92

КБПЗ-2025

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АВІ	– адміністратора віртуальної інфраструктури
АІБ	– адміністратор інформаційної безпеки
ВІ	– віртуальна інфраструктура
ЗЗІ	– засоби захисту інформації
ПД	– персональні дані
ПЗ	– програмне забезпечення
ЦЗОД	– центр зберігання й обробки даних
ЦОД	– центр обробки даних
DLP	– захист від витоків даних
IPS	– системи запобігання вторгнень
CIRC	– Cross Interleaved Reed Solomon Code
EAB	– Embedded Array Block, блок зосередженої пам'яті
ECC	– error-correcting code, код корекції помилок
FEC	– метод прямої корекції помилок
LDPC	– коди Галлагера
NAK	– негативне підтвердження

ВСТУП

Актуальність теми. Мережевий накопичувач (NAS) – це система жорстких дисків, підключена до вашої локальної мережі, а не до окремого комп'ютера. Це дозволяє будь-кому, хто знаходиться у вашому місці розташування, отримувати доступ до файлів, а деякі конфігурації пропонують можливості віддаленого доступу.

Системи NAS зазвичай поєднують кілька дисків, щоб вони виглядали як один великий пристрій зберігання даних. Залежно від конфігурації, вони можуть створювати системи швидше, ніж диски настільних комп'ютерів, або збільшувати резервування даних, щоб збої дисків не руйнували вашу роботу – деякі роблять і те, і інше.

Хоча традиційно вони будуються на фізичних жорстких дисках, сучасні системи NAS підтримують SSD або комбінації SSD/жорсткий диск. Такий гібридний підхід пришвидшує передачу даних, водночас знижуючи витрати, ніж у конфігураціях, що використовують виключно SSD.

Навіть для окремих користувачів NAS забезпечує величезне сховище та резервування даних, що забезпечує спокій. Багато домашніх NAS-систем включають функції потокової передачі медіа, що дозволяє транслювати телебачення та фільми по всій мережі.

Незалежно від того, чи редагуєте ви відео 8K, архівуєте бібліотеки фотографій, чи потребуєте розширюваного безпечного сховища, NAS – це ідеальний центр для творчих робочих процесів.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

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

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Огляд існуючих систем інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

– Дослідження системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

– Програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Об'єктом дослідження є процес інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Предметом дослідження є методи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Методи дослідження базуються на методах теорії великих даних, теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

– Розроблено вітчизняний продукт інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2025

					VKPM-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система NAS – це пристрій зберігання даних, підключений до мережі, який дозволяє зберігати та отримувати дані з централізованого місця для авторизованих користувачів мережі та гетерогенних клієнтів. Ці пристрої зазвичай складаються з механізму, який реалізує файлові служби (пристрій NAS), та одного або кількох пристроїв, на яких зберігаються дані (накопичувачі NAS).

Мета системи NAS – забезпечити локальну мережу (LAN) спільним файловим сховищем у вигляді пристрою, оптимізованого для швидкого зберігання та пошуку даних. NAS – це відносно дорогий варіант зберігання, тому його слід використовувати лише для гарячих даних, до яких звертаються найчастіше. Багато корпоративних ІТ-організацій сьогодні прагнуть перенести дані NAS та об'єктні дані в хмару, щоб зменшити витрати, підвищити гнучкість та ефективність.

1.2 Область застосування

Переваги NAS-сховищ

Мережеві сховища використовуються для зняття відповідальності за обслуговування файлів з інших серверів у мережі та забезпечують зручний спосіб обміну файлами між кількома комп'ютерами. Переваги виділеного мережевого сховища включають:

- Швидший доступ до даних.
- Легко масштабуватися та розширюватися.
- Віддалений доступ до даних.
- Легше адміністрування.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

- Сумісність незалежно від ОС (працює з пристроями на базі Windows та Apple).
- Вбудована безпека даних із сумісністю з резервними масивами зберігання даних.
- Просте налаштування та керування (зазвичай для роботи не потрібен ІТ-фахівець).

Протоколи доступу до файлів NAS

Мережеві пристрої зберігання даних часто здатні взаємодіяти за допомогою різних протоколів доступу до файлів, таких як:

- Мережева файлова система (NFS).
- Блок повідомлень сервера (SMB).
- Протокол подання документів Apple (AFP).
- Загальна файлова система Інтернету (CIFS).

Більшість пристроїв NAS мають гнучкий діапазон систем зберігання даних, з якими вони сумісні, але завжди слід переконатися, що ваш пристрій працюватиме з вашою конкретною системою зберігання даних.

Корпоративні NAS-сховища

У підприємстві масив NAS може використовуватися як основне сховище для зберігання неструктурованих даних та як резервне копіювання для архівування даних або аварійного відновлення (DR). Він також може функціонувати як сервер електронної пошти, бази даних медіа або сервера друку для малого бізнесу. Пристрої NAS вищого класу можуть вмістити достатню кількість дисків для підтримки RAID, технології зберігання даних, яка дозволяє об'єднувати кілька жорстких дисків в один блок, забезпечуючи кращу продуктивність, резервування та високу доступність.

Дані на системах NAS (також відомих як пристрої NAS) часто дзеркально відображаються (реплікуються) на іншу систему NAS, а резервні копії або знімки сліду зберігаються на NAS протягом тижнів або місяців. Це призводить до того, що щонайменше три або більше копій даних зберігаються на дорогому сховищі

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

NAS. Рішення для зберігання даних NAS не потрібно використовувати для аварійного відновлення та резервного копіювання, оскільки це може бути дуже дорого. Знаходячи та розподіляючи дані на рівні (або архівуючи дані) холодних даних з NAS, ви можете усунути зайві копії холодних даних та скоротити витрати на зберігання холодних даних більш ніж на 70%.

Рівні та архівування даних мережевого сховища даних (NAS)

Оскільки сховища NAS зазвичай розроблені для підвищення продуктивності та можуть бути дорогими, дані на NAS часто розподіляються на рівні, архівуються та переміщуються до менш дорогих класів сховищ. Постачальники NAS пропонують деяке базове розподілення даних на рівні блоків, щоб забезпечити обмежену економію на витратах на зберігання, але не на витратах на резервне копіювання та аварійне відновлення. На відміну від власного розподілу на рівні блоків, розподіл або архівування на рівні файлів забезпечує стандартизоване, невласне рішення для максимізації економії шляхом переміщення холодних даних до дешевших рішень для зберігання. Це можна зробити прозоро, щоб користувачі та програми не бачили жодної різниці під час архівування холодних файлів. Прочитайте цей інформаційний документ, щоб дізнатися більше про відмінності між розподілом на рівні файлів та розподілом на рівні блоків .

Міграція NAS до хмари

Хмарні NAS набирають популярності. Але правильний підхід до міграції неструктурованих даних у хмару є надзвичайно важливим. Неструктуровані дані є скрізь. Від геноміки та медичної візуалізації до потокового відео, електромобілів та продуктів Інтернету речей, усі сектори генерують неструктуровані файлові дані. Підприємства, що працюють з великим обсягом даних, зазвичай мають петабайти файлових даних, які можуть складатися з мільярдів файлів, розкиданих по різних постачальниках сховищ, архітектурах та місцях розташування. І хоча зростання файлових даних стрімко зростає, ІТ-бюджети – ні. Саме тому ІТ-організації підприємств прагнуть мігрувати файлові

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

робочі навантаження в хмару. Однак вони стикаються з багатьма перешкодами, через які міграція може тривати тижні або місяці та вимагати значних ручних зусиль.

Проблеми міграції хмарних NAS-серверів

Поширені проблеми міграції неструктурованих даних включають:

– Мільярди файлів, переважно невеликих: Неструктурована міграція даних часто вимагає переміщення мільярдів файлів, переважна більшість з яких є невеликими файлами, що мають величезні накладні витрати, що призводить до повільної передачі даних.

– Протоколи "балаканини": робочі навантаження протоколу блоків повідомлень сервера (SMB), які можуть бути даними користувача, автоматизацією електронного проектування (EDA) та іншими мультимедійними файлами або корпоративними ресурсами, часто є проблемою, оскільки протокол вимагає багатьох обмінів даними, що збільшує трафік у мережі.

– Велика затримка WAN: мережеві файлові протоколи надзвичайно чутливі до мережевих з'єднань з високою затримкою, яких практично неможливо уникнути під час міграції до глобальної мережі (WAN).

– Обмежена пропускна здатність мережі: пропускна здатність часто обмежена або не завжди доступна, що призводить до повільної, ненадійної та складної для керування передачі даних.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Synology DS220+

Дводисковий накопичувач Synology початкового рівня поєднує в собі простоту та можливості. Легке налаштування DSM, чудові сторонні програми (Hyper Backup, Drive, Photos) та тиха робота роблять його ідеальним для творців, яким потрібне надійне резервне копіювання та обслуговування невеликих медіафайлів без складних проблем із налаштуванням.

Переваги:

- Просте налаштування та відшліфований інтерфейс програмного забезпечення DSM.
- Тиха робота та компактний розмір ідеально підходять для домашніх офісів.
- Чудові сторонні програми для резервного копіювання та керування медіафайлами.

Недоліки:

- Обмежено лише двома відсіками, що обмежує потенціал майбутнього зростання..
- Бракує вбудованих мережевих можливостей 2.5GbE або 10GbE.
- Немає слотів NVMe для прискорення кешування SSD.

Специфікації:

- Відсіки: 2 (з можливістю гарячої заміни).
- Процесор: Intel Celeron J4025 (двоядерний).
- Оперативна пам'ять: 2 ГБ DDR4 (з можливістю розширення до 6 ГБ).

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

- Мережа: 2 x 1GbE (агрегація каналів).
- Кеш NVMe/SSD: через відсіки для 2,5-дюймових SSD-накопичувачів (без слотів NVMe).

- Максимальна необроблена ємність: ~36 ТБ (з 2 дисками по 18 ТБ).
- RAID: Hybrid RAID від Synology, RAID 0/1, JBOD.
- Розміри: 165 x 108 x 232 мм (6,5 x 4,3 x 9,1 дюйма).
- Вага: 1,3 кг (2,9 фунта).

Призначення:

- Початківці та творці-одинаки, яким потрібні надійні резервні копії, бібліотеки фотографій та прості у використанні рішення, такі як Plex або Time Machine.

ТерраМастер F2-423

Бюджетний потужний накопичувач TerraMaster поєднує вражаючі характеристики в доступному корпусі. Завдяки двом портам 2,5 Гбіт/с, двом слотам M.2 NVMe SSD та 4 ГБ оперативної пам'яті з можливістю розширення, F2-423 пропонує чудове співвідношення ціни та якості для творців, яким потрібна швидша мережа та кешування SSD без преміальної ціни.

Переваги:

- Два порти 2.5GbE для швидшого мережевого з'єднання.
- Два слоти M.2 NVMe для кешування SSD або додаткового сховища.
- Відмінне співвідношення ціни та якості обладнання з 4 ГБ оперативної пам'яті та чотирьохядерним процесором.

Недоліки:

- Операційна система TOS менш відшліфована, ніж інтерфейс DSM від Synology.
- Якість збірки виглядає бюджетною порівняно з преміальними альтернативами.
- Менша екосистема додатків, ніж у відомих конкурентів.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Специфікації:

- Відсіки: 2 (з можливістю гарячої заміни).
- Процесор: Intel Celeron N5105 (чотириядерний).
- Оперативна пам'ять: 4 ГБ DDR4 (з можливістю розширення до 32 ГБ).
- Мережа: 2 x 2.5GbE (агрегація каналів до 5 Гбіт/с).
- Кеш NVMe/SSD: 2 слоти M.2 NVMe (PCIe 3.0).
- Максимальна необроблена ємність: ~40 ТБ (з 2 дисками по 20 ТБ).
- RAID: 0/1, JBOD, одинарний, TRAIID.
- Розміри: 158 x 225 x 142 мм (6,2 x 8,9 x 5,6 дюйма).
- Вага: 1,5 кг (3,3 фунта).

Призначення:

- Бюджетні творці, які бажають потужного обладнання та швидшої мережі, не сплачуючи преміальних цін.

Asustor AS5304T

Завдяки чотирьом відсікам для гарячої заміни та двом портам 2.5GbE, AS5304T – це зручний для розвитку NAS-сервер для домашніх офісів та початківців у сфері відео. Якщо вам подобається потокове передавання медіа, він достатньо швидкий для багатопотокового 4K Plex, а магазин додатків ADM охоплює резервне копіювання, хмарну синхронізацію та програмне забезпечення для відеоспостереження.

Переваги:

- Два порти 2.5GbE забезпечують швидшу передачу даних, ніж стандартні гігабітні мережеві пристрої.
- Чотири відсіки забезпечують належне резервування RAID та майбутнє розширення.
- Потужні можливості транскодування медіа для потокової передачі контенту 4K.

Недоліки:

- Вентилятори можуть бути голоснішими під час високого навантаження.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

– Інтерфейс програмного забезпечення не такий відшліфований, як у інших брендів.

Специфікації:

- Відсіки: 4 (з можливістю гарячої заміни).
- Процесор: Intel Celeron J4105 (чотириядерний).
- Оперативна пам'ять: 4 ГБ DDR4 (розширюється користувачем).
- Мережа: 2 x 2.5GbE (агрегація каналів до 5 Гбіт/с).
- Кеш NVMe/SSD: через відсіки для 2,5-дюймових SSD-накопичувачів (без слотів NVMe).

- Максимальна необроблена ємність: ~72 ТБ (з 4 дисками по 18 ТБ).
- RAID: 0/1/5/6/10, JBOD, одинарний.
- Розміри: 170 x 174 x 230 мм (6,7 x 6,8 x 9,1 дюйма).
- Вага: 2,2 кг (4,9 фунта).

Призначення:

– Користувачі, яким потрібен вигідний 4-відсіковий комп'ютер з мережею швидше за гігабіт та можливостями для зростання.

QNAP TS-464

TS-464 від QNAP поєднує гнучкість у компактному 4-відсіковому дизайні, оснащеному двома портами 2,5 Гбіт/с для швидкої передачі даних, двома слотами NVMe для кешування та слотом PCIe для додаткових портів, таких як 10 Гбіт/с або додаткові порти USB. Творці також можуть використовувати вихід HDMI для прямого підключення до Plex box або попереднього перегляду на місці без необхідності використання окремого ПК.

Переваги:

- Два порти 2.5GbE зі слотом розширення PCIe для майбутніх оновлень.
- Два слоти M.2 NVMe забезпечують високошвидкісне кешування SSD.
- Вихід HDMI дозволяє пряме підключення до дисплеїв для потокової передачі медіа.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Недоліки:

- Інтерфейс користувача QTS складніший, ніж у інших брендів.
- Шум вентилятора може бути помітний під час високого навантаження на процесор.
- Крива навчання крутіша для новачків у NAS-системах.

Специфікації:

- Відсіки: 4 (з можливістю гарячої заміни).
- Процесор: Intel Celeron N5105 (чотириядерний).
- Оперативна пам'ять: 4 ГБ DDR4 (з можливістю розширення до 16 ГБ).
- Мережа: 2 x 2.5GbE.
- Кеш NVMe/SSD: 2 слоти M.2 NVMe (кеш або пули сховищ).
- Розширення: слот PCIe Gen3 x2 (10GbE, USB тощо).
- Відео: HDMI 2.0 (4K60).
- Максимальна необроблена ємність: ~72 ТБ (з 4 дисками по 18 ТБ).
- RAID: 0/1/5/6/10, JBOD.
- Розміри: 168 x 170 x 226 мм (6,6 x 6,7 x 8,9 дюйма).
- Вага: 2,3 кг (5,1 фунта).

Призначення:

Творці, що розвиваються, та невеликі команди, яким потрібна швидкість зараз, а можливості для оновлення – пізніше (10GbE, кеш NVMe, Plex з прямим підключенням до телевізора).

Synology DS224+

DS224+ поєднує в собі вдосконалене програмне забезпечення Synology DSM у потужному дводисковому комп'ютері. Завдяки чотириядерному процесору Intel, можливостям перекодування відео та двом гігабітним портам, він ідеально підходить для творців, яким потрібне високоякісне програмне забезпечення Synology з достатньою потужністю для Plex та легкої віртуалізації.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Переваги:

- Відшліфований інтерфейс DSM з чудовою екосистемою додатків та мобільними клієнтами.
- Підтримка апаратного перекодування для потокової передачі відео 4K через Plex.
- Два гігабітні порти Ethernet забезпечують агрегацію каналів для підвищення швидкості для домашніх та студійних користувачів.

Недоліки:

- Обмежено двома відсіками для дисків, що обмежує можливості розширення.
- Немає вбудованого мережевого зв'язку 2.5GbE або 10GbE, як у новіших альтернатив.
- Відсутні слоти M.2 NVMe для прискорення кешування SSD.

Специфікації:

- Відсіки: 2 (з можливістю гарячої заміни).
- Процесор: Intel Celeron J4125 (чотириядерний 2,0 ГГц, з можливістю розгону до 2,7 ГГц).
- Оперативна пам'ять: 2 ГБ DDR4 (з можливістю розширення до 6 ГБ).
- Мережа: 2 x 1GbE (агрегація каналів).
- Кеш NVMe/SSD: через відсіки для 2,5-дюймових SSD-накопичувачів (без слотів NVMe).
- Максимальна необроблена ємність: ~40 ТБ (з 2 дисками по 20 ТБ).
- RAID: SHR, RAID 0/1, JBOD.
- Розміри: 165 x 108 x 232 мм (6,5 x 4,3 x 9,1 дюйма).
- Вага: 1,3 кг (2,9 фунта).

Призначення:

- Шанувальники Synology, яким потрібні можливості транскодування та перевірена надійність програмного забезпечення в компактному форм-факторі з 2 відсіками.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Synology DS923+

DS923+ пропонує фотографам найкращу 4-слотову платформу Synology, оснащену швидким процесором Ryzen, до 32 ГБ оперативної пам'яті, двома слотами NVMe (тепер можна використовувати для зберігання даних з DSM 7.2) та додатковим підключенням 10GbE через додаткову карту. Поєднайте його із Synology Photos та Hyper Backup для відшліфованого, інтегрованого робочого процесу, який масштабується.

Переваги:

- Відшліфований інтерфейс DSM з комплексними інструментами резервного копіювання та керування фотографіями.
- Два слоти M.2 NVMe підтримують як кешування, так і пули сховищ.
- Розширювана оперативна пам'ять до 32 ГБ забезпечує розширену віртуалізацію та високі робочі навантаження.

Недоліки:

- Тільки мережевий зв'язок 1GbE на борту, потрібна додаткова карта для швидкості 10GbE.
- Дорожче, ніж у конкурентів з аналогічними характеристиками обладнання.
- Розширення eSATA обмежене порівняно з новішими варіантами підключення.

Специфікації:

- Відсіки: 4 (з можливістю гарячої заміни).
- Процесор: AMD Ryzen R1600 (двоядерний з SMT).
- Оперативна пам'ять: 4 ГБ DDR4 ECC (з можливістю розширення до 32 ГБ).
- Мережа: 2 x 1GbE (додатково E10G22-T1-Mini для 10GbE).
- NVMe/SSD: 2 слоти M.2 NVMe (кеш або сховище).
- Розширення: eSATA для розширення Dх517 (до 9 відсіків загалом).
- Максимальна необроблена ємність: ~108 ТБ (з 4 дисками по 27 ТБ).

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

- RAID: SHR, RAID 0/1/5/6/10, JBOD.
- Розміри: 166 x 199 x 223 мм (6,5 x 7,8 x 8,8 дюйма).
- Вага: 2,24 кг (4,9 фунта).

Призначення:

- Фотографи та творці гібридного контенту, яким потрібне програмне забезпечення Synology з масштабуванням професійного рівня та опціональною підтримкою 10GbE.

Jellyfish Nomad OWC

Jellyfish Nomad переосмислює портативні NAS-системи для професійних відеороботів. Завдяки 64 ТБ повністю NVMe-сховища, стабільній швидкості 6000 МБ/с та шістьом портам 10GbE, він розроблений для DIT, студій візуальних ефектів та виїзних команд, яким потрібна продуктивність серверного класу у форм-факторі ручної поклажі. Створений відеопрофесіоналами для відеопрофесіоналів, він забезпечує корпоративні можливості без зайвої складності.

Переваги:

- Виняткова стабільна швидкість читання/запису 6000 МБ/с завдяки повністю NVMe-архітектурі сховища.
- Портативний дизайн, сумісність з ручною поклажею авіакомпаній та м'який чохол у комплекті.
- Шість портів 10GbE забезпечують пряме підключення кількох високопродуктивних робочих станцій.

Недоліки:

- Ціни преміум-класу значно вищі, ніж у традиційних настільних NAS-рішень.
- Обмежено відеоорієнтованими робочими процесами та професійним виробничим середовищем.
- Потрібна високошвидкісна мережева інфраструктура для повного використання можливостей продуктивності.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

– Файлова система QuTS Hero на базі ZFS забезпечує функції цілісності корпоративних даних.

Недоліки:

– Великий розмір та високе енергоспоживання не підходять для домашнього офісу.

– Преміум-ціни роблять його недоступним для звичайних користувачів.

– Складний набір функцій вимагає технічної експертизи для повного використання.

Специфікації:

– Відсіки: 8 (з можливістю гарячої заміни).

– Процесор: варіанти Intel Core i5/i7/i9 (для настільних комп'ютерів).

– Оперативна пам'ять: 16 ГБ DDR4 (з можливістю розширення до 128 ГБ).

– Мережа: 2 x 2.5GbE; слоти PCIe для 10/25GbE.

– NVMe/SSD: слоти M.2 NVMe (залежно від моделі).

– Розширення: Кілька слотів PCIe (графічні процесори, мережеві карти, QM2).

– Максимальна необроблена ємність: ~160 ТБ+ (з 8 дисками по 20 ТБ).

– Файлова система: QuTS hero (ZFS) зі знімками та дедуплікацією даних.

– Розміри: 188 x 329 x 279 мм (7,4 x 12,9 x 11,0 дюймів).

– Вага: 8,0 кг (17,6 фунта).

Призначення:

– Студії, яким потрібна пропускна здатність робочого місця, цілісність даних та розширення для відеоконвеєрів з кількома редакторами.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення написано мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

орієнтована мова, призначена для розробки різноманітних безпечних і потужних додатків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські додатки Windows, веб-служби XML, розподілені компоненти, додатки типу “ сервер-клієнт”, додатки баз даних і багато яких інших. В Visual C# 2012 є розширений редактор коду, конструктори зі зручним користувальницьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку додатків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

Синтаксис Visual C# дуже виразний, але простий у вивченні. Усі, хто знаком з мовами C, C++ або Java з легкістю визнають синтаксис із фігурними дужками, характерний для мови Visual C#. Розроблювачі, що знають кожен із цих мов, як правило, зможуть домогтися ефективної роботи з мовою Visual C# за дуже короткий час. Синтаксис Visual C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вираження й прямий доступ до пам'яті, чого немає в Java. Visual C# підтримує універсальні методи й типи, забезпечуючи більше високий рівень безпеки й продуктивності, а також ітератори, що дозволяють при реалізації колекцій класів визначати власне поводження ітерації, що може легко використовуватися в клієнтському коді. В Visual C# 5.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласною конструкцією мови.

Як об'єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод Main – точку входу додатка – інкапсулюється у визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що

розподіляється по стопках, що реалізує інтерфейси, але не підтримує спадкування.

На додаток до основних описаних об'єктно-орієнтованих принципів, мова Visual C# спрощує розробку компонентів програмного забезпечення завдяки декільком інноваційним конструкціям мови, у число яких входять наступні:

- Інкапсульовані підписи методів, називані делегатами, які підтримують строго-типізовані повідомлення про події.
- Властивості, що виступають у ролі методів доступу для закритих змінних-членів.
- Атрибути з декларативними метаданими про типи під час виконання.
- Вбудовані коментарі XML-документації.
- LINQ (Language-Integrated Query), що пропонує вбудовані можливості запитів у різних джерелах даних.

Якщо буде потрібно забезпечити взаємодію з іншим програмним забезпеченням Windows, таким як об'єкти COM або власні бібліотеки DLL Win32, у мові Visual C# можна використовувати процес, що називається "Interop". Процес Interop дозволяє програмам на Visual C# виконувати практично будь-які дії, які може виконувати вихідний додаток на C++. Мова Visual C# підтримує навіть покажчики й поняття "небезпечного" коду для тих випадків, коли прямий доступ до пам'яті має вкрай важливе значення.

Процес побудови Visual C# у порівнянні з C і C++ простий і є більше гнучким, чим в Java. Немає окремих файлів заголовка, а методи й типи не потрібно повідомляти в певному порядку. У вихідному файлі Visual C# може бути визначене будь-яке число класів, структур, інтерфейсів і подій.

Архітектура платформи .NET Framework

Програма мовою Visual C# виконується в середовищі .NET Framework – інтегрованому компоненті Windows, що містить віртуальну систему виконання (середовище CLR) і уніфікований набір бібліотек класів. Середовище CLR являє собою комерційну реалізацію корпорацією Майкрософт інфраструктури CLI, що

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

є міжнародним стандартом, який лежить в основі створення середовищ виконання й розробки, у яких забезпечується тісна взаємодія між мовами й бібліотеками.

Вихідний код, написаний мовою Visual C#, компілюється в проміжну мову (IL) у відповідності зі специфікацією CLI. Код IL і ресурси, такі як растрові зображення й рядки, зберігаються на диску у файлі, що виконується, названому складанням, з розширенням EXE або DLL у більшості випадків. Складання містить маніфест із відомостями про типи складання, версії, мови й регіональні параметри та вимоги безпеки.

При виконанні програми на Visual C# складання завантажується в середовище CLR залежно від відомостей у маніфесті. Далі, якщо вимоги безпеки дотримані, середовище CLR виконує JIT-компіляцію для перетворення коду IL в інструкції машинного коду. Середовище CLR також надає інші служби, що відносяться до автоматичного збору сміття, обробки виключень і керуванню ресурсами. Код, виконуваний середовищем CLR, іноді називають "керованим кодом" у протиставлення "некерованому коду", що компілюється в машинний код, призначений для певної системи. Далі показані відносини під час компіляції й час виконання між файлами з вихідним кодом Visual C#, бібліотеками класів .NET Framework, складаннями й середовищем CLR.

Взаємодія між мовами є ключовою особливістю .NET Framework. Оскільки код IL, створюваний компілятором Visual C# відповідає Специфікації: CTS, код IL на основі Visual C# може взаємодіяти з кодом, створюваним версіями мов Visual Basic, Visual C++, Visual J# платформи .NET Framework і ще більш ніж 20 CTS-сумісних мов. В одному складанні може бути кілька модулів, написаних на різних мовах платформи .NET Framework, і типи можуть посилатися один на одного, як якби вони були написані на одній мові.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) 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.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розуміння різних типів мережевих сховищ є важливим для ефективного керування даними. У цій статті буде розглянуто різні типи мережевих сховищ, включаючи мережеві сховища (NAS), мережі зберігання даних (SAN), безпосередньо підключені сховища (DAS), хмарні сховища, розподілені файлові системи та об'єктні сховища. У цьому розділі розглядаються ці типи мережевих сховищ, а також надається уявлення про їхні функції, переваги та застосування, щоб допомогти вам приймати обґрунтовані рішення, адаптовані до ваших потреб в управлінні даними.

Типи мережевих сховищ відіграють вирішальну роль у сучасній ІТ-інфраструктурі, надаючи організаціям масштабовані та ефективні стратегії управління даними. Мережі зберігання даних (SAN), мережеві сховища даних (NAS) та сховища даних з прямим підключенням (DAS) представляють собою різні підходи до зберігання та доступу до даних, кожен з яких має унікальні характеристики, адаптовані до різних операційних потреб.

SAN пропонують високошвидкісне сховище на рівні блоків, яке зазвичай використовується в корпоративних середовищах, що вимагають високої продуктивності та централізованого управління сховищем.

Системи NAS забезпечують сховище на рівні файлів, доступне через мережеві протоколи, що робить їх ідеальними для спільних файлових середовищ та спільних робочих просторів. На відміну від них, DAS підключає пристрої зберігання даних безпосередньо до певного сервера або комп'ютера, пропонуючи простіше та економічніше рішення для потреб меншого обсягу сховища.

Ці технології зберігання відрізняються своєю архітектурою, доступністю, масштабованістю та продуктивністю, що дозволяє ІТ-фахівцям вибирати

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

найбільш підходяще рішення на основі конкретних вимог організації до зберігання даних, продуктивності та доступності.

Розуміння мережевого сховища (NAS)

Мережеве сховище даних (NAS) – це спеціальний тип сховища, яке підключається до мережі. Воно надає багатьом користувачам і пристроям простий спосіб зберігання, доступу та обміну файлами. Це робить його чудовим варіантом як для домашнього, так і для комерційного використання.

Переваги NAS:

- Легкий доступ: Користувачі можуть отримувати доступ до файлів з будь-якого пристрою, підключеного до мережі.
- Централізоване сховище: усі файли зберігаються в одному місці, що спрощує керування ними.
- Економічно ефективні: NAS-системи, як правило, дешевші за інші рішення для зберігання даних.

Обмеження NAS:

- Проблеми зі швидкістю: Якщо забагато користувачів одночасно отримують доступ до NAS, це може призвести до уповільнення його роботи.
- Залежність від мережі: якщо мережа виходить з ладу, доступ до файлів втрачається.
- Обмежена продуктивність: Не така швидка, як у сховищ із прямим підключенням (DAS) для важких завдань.

Випадки використання NAS:

- Домашнє використання: Люди використовують NAS для зберігання фотографій, відео та музики.
- Малий бізнес: Це допомагає в обміні документами та резервному копіюванні даних.
- Віддалена робота: команди можуть легко отримувати доступ до файлів з різних місць.

Таким чином, NAS – це гнучкий та ефективний спосіб керування даними

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

для різних потреб, що робить його популярним вибором для багатьох користувачів.

Дослідження мереж зберігання зон (SAN)

Огляд архітектури SAN

Мережа зберігання даних (SAN) – це спеціальний тип мережі, яка з'єднує пристрої зберігання даних із серверами. Така конфігурація дозволяє кільком серверам отримувати доступ до спільного сховища, ніби це локальний диск. SAN призначені для ефективної обробки великих обсягів даних, що робить їх важливими для підприємств, яким потрібен швидкий доступ до інформації.

Переваги SAN:

- Висока продуктивність: мережі SAN забезпечують високу швидкість передачі даних, що є критично важливим для програм, яким потрібен швидкий доступ до великих файлів.
- Масштабованість: компанії можуть легко додавати більше пристроїв зберігання даних до мережі зберігання даних, не порушуючи існуючі операції.
- Централізоване управління: мережі SAN дозволяють легше керувати ресурсами зберігання даних, що спрощує розподіл простору та моніторинг їх використання.

SAN проти NAS

Підсумовуючи, хоча SAN та NAS пропонують рішення для зберігання даних, вони задовольняють різні потреби. SAN ідеально підходять для високопродуктивних середовищ, тоді як NAS більше підходить для загального зберігання та обміну файлами.

Огляд безпосередньо підключених сховищ (DAS)

Пряме підключене сховище (DAS) являє собою традиційну архітектуру сховища, де пристрої зберігання даних безпосередньо підключені до одного комп'ютера або сервера через інтерфейси, такі як SATA, SCSI або SAS. На відміну від мережеских рішень для зберігання даних, DAS фізично підключений до хост-системи, забезпечуючи просте та безпосереднє підключення до сховища.

					ВКРМ-122.25.0043.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		27

Варіанти використання:

- Малий бізнес з обмеженими вимогами до зберігання.
- Індивідуальні робочі станції та персональні обчислювальні середовища.
- Сценарії, що вимагають високошвидкісних, локалізованих рішень для зберігання даних.
- Системи резервного копіювання та архівування з мінімальними потребами в спільному доступі.

Порівняння з мережевим сховищем

Хоча DAS пропонує простоту та безпосередню продуктивність, йому бракує гнучкості та масштабованості мережевих рішень для зберігання даних, таких як SAN та NAS, що робить його більш придатним для конкретних, локалізованих сценаріїв зберігання даних.

Отже, який тип сховища найкраще підходить для моєї мережі?

Тип мережі, який ви оберете, залежатиме, перш за все, від найкращого рішення для зберігання даних для вашого бізнесу. Після того, як ви визначите фактори, які є вирішальними для вашої діяльності, ви зможете вибрати варіант зберігання, який відповідає вашим конкретним потребам.

3.2 Розробка структурної схеми

Хмарне сховище – це хмарний сервіс обчислень, який дозволяє користувачам зберігати дані та файли поза межами свого офісу у стороннього постачальника. Це означає, що ви можете отримати доступ до своїх файлів через Інтернет, що полегшує отримання інформації з будь-якого місця. Ось деякі важливі аспекти хмарного сховища:

Види хмарних сховищ:

- Публічна хмара: такі сервіси, як Google Диск і Dropbox, пропонують сховище, до якого може отримати доступ кожен.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

– Приватна хмара: вона призначена для однієї організації, що забезпечує більший контроль та безпеку.

– Гібридна хмара: інтегрує публічні та приватні хмари, забезпечуючи гнучкість в управлінні даними.

Безпека хмарного сховища:

– Шифрування: дані кодуються для захисту від несанкціонованого доступу.

– Контроль доступу: Тільки авторизовані користувачі мають доступ до певних файлів.

– Регулярне резервне копіювання: гарантує, що дані не будуть втрачені у разі збою.

Підсумовуючи, хмарні рішення для зберігання даних пропонують зручний спосіб керування даними без потреби у фізичних пристроях зберігання. Вони надають різні опції для задоволення різних потреб, гарантуючи, що користувачі можуть безпечно та ефективно отримувати доступ до своїх даних.

Розподілені файлові системи

Розподілені файлові системи (DFS) призначені для керування файлами на кількох серверах, що полегшує користувачам доступ до даних з різних місць. Ці системи є важливими для організацій, яким потрібно ефективно обмінюватися великими обсягами даних.

Як працюють розподілені файлові системи

Розподілені файлові системи працюють, розбиваючи файли на менші частини та зберігаючи їх на різних серверах. Це дозволяє:

– Покращена продуктивність: кілька серверів можуть обробляти запити одночасно.

– Масштабованість: можна додавати більше серверів у міру зростання потреб у даних.

– Відмовостійкість: якщо один сервер вийде з ладу, інші все ще можуть забезпечити доступ до даних.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Економічна ефективність: Зберігання великих обсягів даних часто коштує менше порівняно з традиційними методами.

– Доступність: До даних у сховищі об'єктів можна отримати доступ з будь-якого місця, що робить його ідеальним для віддаленої роботи.

Застосування використання зберігання об'єктів:

– Медіа та розваги: Зберігання великих відеофайлів та зображень.

– Охорона здоров'я: керування записами пацієнтів та даними візуалізації.

– Резервне копіювання та архівування: безпечне зберігання копій важливих даних.

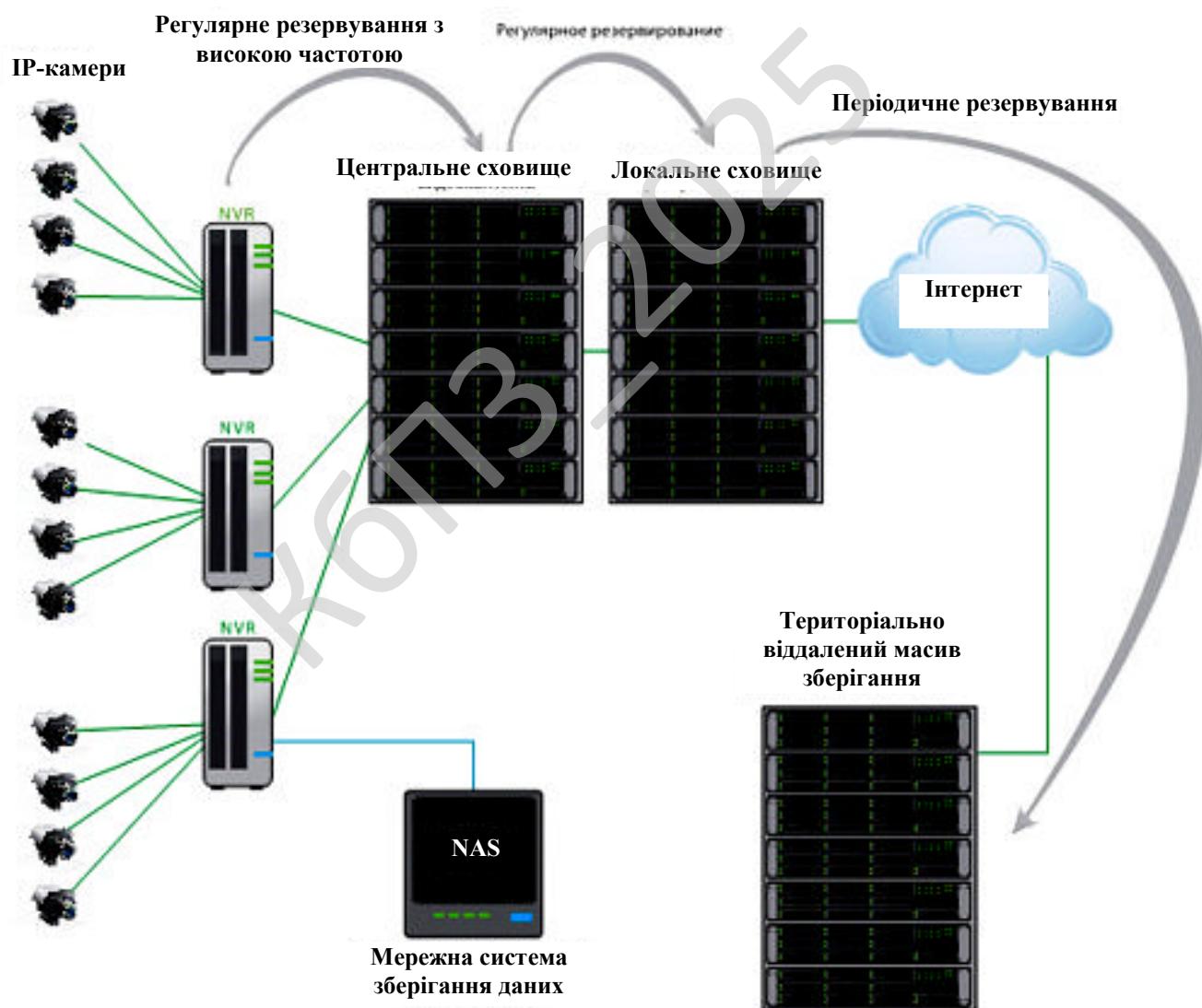


Рисунок 3.1 – Структурна схема системи

Підсумовуючи, хоча традиційні методи зберігання даних, такі як NAS та SAN, мають своє місце, об'єктне сховище пропонує унікальні переваги, що роблять його придатним для сучасних потреб у даних. Оскільки бізнес продовжує генерувати більше неструктурованих даних, роль об'єктного сховища, ймовірно, зростатиме ще більшою.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. З неї бачимо, що розроблене програмне забезпечення системи підвищення надійності зберігання даних на сервері складається з наступних основних функціональних блоків:

- сам носій інформації – сервери мережних систем зберігання;
- кодер Ріда-Соломона, який використовується, коли відбувається запис інформації на сервер, при цьому необхідно враховувати, що об'єм інформації, яка записується на диск серверу, повинна бути меншою, ніж об'єм диску, в зв'язку, з тим, що коди Ріда-Соломона відносяться до кодів з надмірністю, за рахунок якої й відбувається кодування;
- декодер Ріда-Соломона, який використовується при читанні даних з відповідного диску.

Функціонально блок, який утримує кодер Ріда-Соломона включає в себе наступні блоки:

- дані, які потрібно зберегти на сервері мережних систем зберігання;
- поліном для кодування;
- відмовостійки коди.

Коди Ріда-Соломона базуються на спеціальному розділі математики – полях Галуа (GF) або кінцевих полях. Арифметичні дії (+, -, x, / і т.д.) над елементами кінцевого поля дають результат, що також є елементом цього поля. Кодер та декодер Ріда-Соломона повинні вміти виконувати ці арифметичні

операції. Ці операції для своєї реалізації вимагають спеціального устаткування або спеціалізованого програмного забезпечення.

Кодове слово Ріда-Соломона формується із залученням спеціального полінома. Всі коректні кодові слова повинні ділитися без залишку на ці утворюючі поліноми. Загальна форма утворюючого полінома має вигляд: $g(x) = (x-a^i)(x-a^{i+1})\dots(x-a^{i+2t})$, а кодове слово формується за допомогою операції: $c(x) = g(x) \cdot i(x)$, де $g(x)$ є утворюючим поліномом, $i(x)$ являє собою інформаційний блок, $c(x)$ – кодове слово, що називається простим елементом поля.

$2t$ символів парності в кодовому слові Ріда-Соломона визначаються з наступного співвідношення: $p(x) = i(x) \cdot x^{n-k} \bmod g(x)$.

Застосування

У даний момент коди Ріда-Соломона мають дуже широку область застосування завдяки їхній здатності знаходити й виправляти багаторазові пакети помилок.

Код Ріда-Соломона використовується при записі й читанні в контролерах оперативної пам'яті, при архівуванні даних, запису інформації на жорсткі диски (ЕСС), запису на CD/DVD диски.

Навіть якщо ушкоджено значний обсяг інформації, зіпсовано кілька секторів дискового носія, то коди Ріда-Соломона дозволяють відновити більшу частину загубленої інформації. Також використовується при записі на такі носії, як магнітні стрічки й штрихкоди.

Можливі помилки при читанні з диска з'являються вже на етапі виробництва диска, тому що зробити ідеальний диск при сучасних технологіях неможливо.

Так само помилки можуть бути викликані подряпинами на поверхні диска, пилом і т.д. Тому при виготовленні компакт-диску, що читається, використовується система корекції CIRC (Cross Interleaved Reed Solomon Code). Ця корекція реалізована у всіх пристроях, що дозволяють зчитувати дані з CD дисків, у вигляді чипа із прошиванням firmware. Знаходження й корекція помилок заснована надмірності й перемеженні (redundancy & interleaving).

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Надмірність приблизно 25% від вихідної інформації.

При записі на цифрові аудіокомпакт-диски (Compact Disc Digital Audio – CD-DA) використовується стандарт Red Book.

Корекція помилок відбувається на двох рівнях C1 і C2.

При кодуванні на першому етапі відбувається додавання перевірочних символів до вихідних даних, на другому етапі інформація знову кодується.

Крім кодування здійснюється також перемішування (перемеження) байтів, щоб при корекції блоки помилок розпалися на окремі біти, які легше виправляються.

На першому рівні виявляються й виправляються помилкові блоки довжиною один і два байти (один і два помилкових символи відповідно). Помилкові блоки довжиною три байти виявляються й передаються на наступний рівень.

На другому рівні виявляються й виправляються помилкові блоки, що виникли в C2, довжиною 1 і 2 байти. Виявлення трьох помилкових символу є фатальною помилкою, не можуть бути виправлені.

Цей алгоритм кодування використовується при передачі даних по мережах WiMAX, в оптичних лініях зв'язку, у супутниковому й радіорелейному зв'язку. Метод прямої корекції помилок у минаючому трафіку (Forward Error Correction, FEC) ґрунтується на кодах Ріда-Соломона.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

Перейдемо до розгляду іншого функціонального блоку – декодеру Ріда-Соломона.

Цей функціональний блок містить у собі наступні блоки:

- відмовостійкий код;
- блок обчислення поліномів синдрому та помилок;
- блок обчислення локації та значень помилок;
- блок корекції помилок;
- відновлені за допомогою кодів Ріда-Соломона дані.



Сервер у мережній системі зберігання

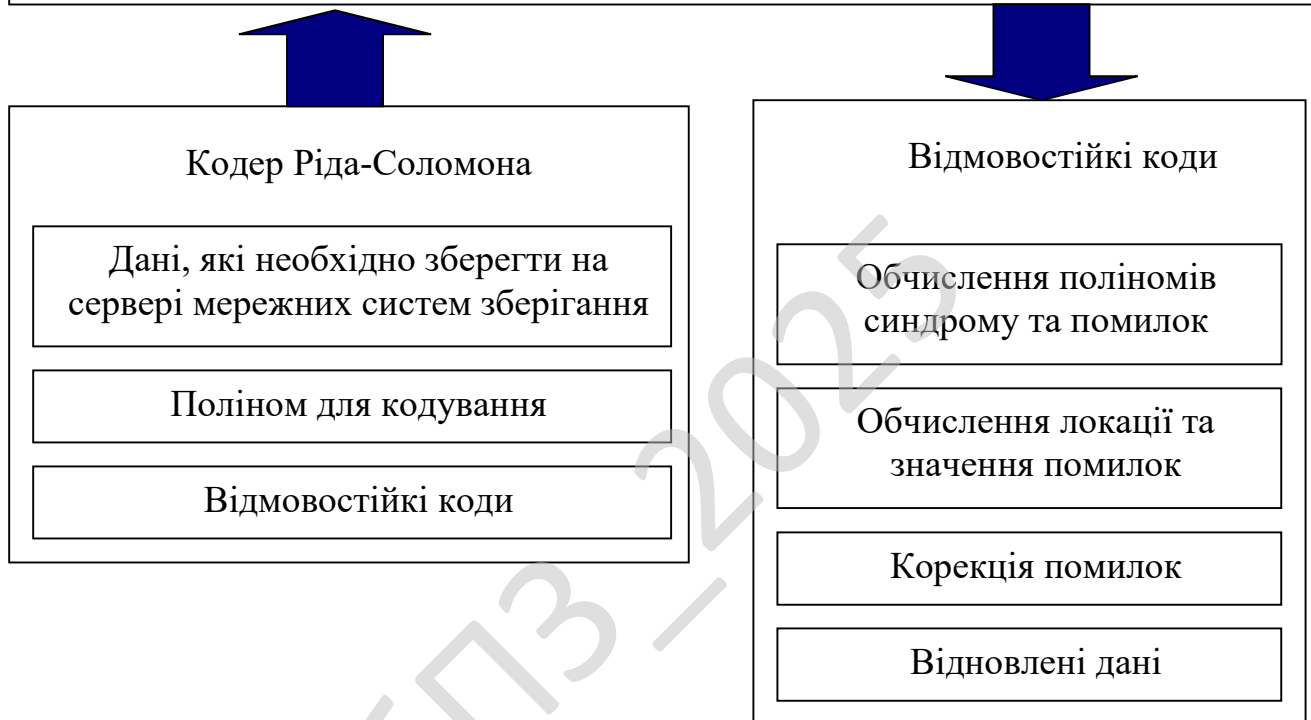


Рисунок 3.2 – Функціональна схема системи

Декодер працює наступним чином.

Введемо позначення:

- $r(x)$ – Отримане кодове слово.
- S_i – Синдроми.
- $L(x)$ – Поліном локації помилок.
- X_i – Положення помилок.
- Y_i – Значення помилок.
- $c(x)$ – Відновлене кодове слово.

– v – Число помилок.

Отримане кодове слово $r(x)$ являє собою вихідне (передане) кодове слово $c(x)$ плюс помилки: $r(x) = c(x) + e(x)$.

Декодер Ріда-Соломона намагається визначити позицію й значення помилки для числа t помилок (або $2t$ втрат) і виправити помилки й втрати.

Обчислення синдрому

Обчислення синдрому схоже на обчислення парності. Кодове слово Ріда-Соломона має $2t$ синдромів, це залежить тільки від помилок (а не переданих кодових слів). Синдроми можуть бути обчислені шляхом підстановки $2t$ коріння утворюючого полінома $g(x)$ в $r(x)$.

Знаходження позицій символічних помилок

Це робиться шляхом рішення системи рівнянь із t невідомими. Існує кілька швидких алгоритмів для рішення цього завдання. Ці алгоритми використовують особливості структури матриці кодів Ріда-Соломона й сильно скорочують необхідну обчислювальну потужність. Робиться це у два етапи:

1. Визначення полінома локації помилок. Це може бути зроблене за допомогою алгоритму Berlekamp-Massey або алгоритму Евкліда. Алгоритм Евкліда використовується частіше на практиці, тому що його легше реалізувати, однак, алгоритм Berlekamp-Massey дозволяє одержати більш ефективну реалізацію встаткування й програм.

2. Знаходження кореня цього полінома. Це робиться із залученням алгоритму пошуку Chien.

Знаходження значень символічних помилок

Тут також потрібно вирішити систему рівнянь із t невідомими. Для рішення використовується швидкий алгоритм Forney.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

Код Ріда-Соломона

Для реалізації перешкодостійкого зберігання інформації у мережних

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

то його порядок дорівнює $q-1$, т.е. $\alpha^{q-1}=1, \alpha^i \neq 1, 0 < i < q-1$.

Тоді нормований поліном $g(x)$ мінімального ступеня над полем $GF(q)$, коріннями якого є $d-1$ ступенів, що йдуть підряд, $\alpha^{l_0}, \alpha^{l_0+1}, \dots, \alpha^{l_0+d-1}$ елемента α , є поліномом, що породжує, коду над полем $GF(q)$ $g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0+1}) \dots (x - \alpha^{l_0+d-1})$, де l_0 – деяке ціле число (у тому числі 0 і 1), за допомогою якого іноді вдається спростити кодер. Звичайно покладається $l_0 = 1$. Ступінь багаточлена $g(x)$ дорівнює $d-1$.

Довжина отриманого коду n , мінімальна відстань d (мінімальна відстань d лінійного коду є мінімальним із всіх відстаней Хеммінга всіх пар кодових слів). Код містить $r = d-1 = \deg(g(x))$ перевірочний символ, де $\deg()$ позначає ступінь полінома; число інформаційних символів $k = n - r = n - d + 1$. У такий спосіб $d = n - k - 1$ і код Ріда-Соломона є роздільним кодом з максимальною відстанню (є оптимальним у змісті границі Синглтона).

Кодовий поліном $c(x)$ може бути отриманий з інформаційного полінома $m(x)$, $\deg m(x) \leq k-1$, шляхом перемножування $m(x)$ і $g(x)$: $c(x) = m(x)g(x)$

Властивості

Код Ріда-Соломона над $GF(q^m)$, що виправляє t помилок, вимагає $2t$ перевірочних символів і з його допомогою виправляються довільні пакети довжиною t і менше. Відповідно до теореми про границю Рейгера, коди Ріда-Соломона є оптимальними з погляду співвідношення довжини пакета й можливості виправлення помилок – використовуючи $2t$ додаткових перевірочних символів виправляються t помилок (і менш).

Теорема (границя Рейгера). Кожний лінійний блоковий код, що виправляє всі пакети довжиною t і менш, повинен містити щонайменше $2t$ перевірочних символів.

Виправлення багаторазових помилок

Код Ріда-Соломона є одним з найбільш потужних кодів, що виправляють багаторазові пакети помилок. Застосовується в каналах, де пакети помилок можуть утворюватися настільки часто, що їх уже не можна виправляти за

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

допомогою кодів, що виправляють одиночні помилки. $(q^m - 1, q^m - 1 - 2t)$ -код Ріда-Соломона над полем $GF(q^m)$ з кодовою відстанню $d = 2t + 1$ можна розглядати як $((q^m - 1)m, (q^m - 1 - 2t)m)$ -код над полем $GF(q)$, що може виправляти будь-яку комбінацію помилок, зосереджену в t або меншому числі блоків з m символів.

Найбільше число блоків довжини m , які може торкнутися пакет довжини l_i , де $l_i \leq mt_i - (m - 1)$, не перевершує t_i , тому код, що може виправити t блоків помилок, завжди може виправити й будь-яку комбінацію з r пакетів загальної довжини l , якщо $l + (m - 1) \leq mt$.

Практична реалізація

Кодування за допомогою коду Ріда-Соломона може бути реалізовано двома способами: систематичним і несистематичним.

При несистематичному кодуванні інформаційне слово множиться на якийсь полином, що неприводиться, у полі Галуа. Отримане закодоване слово повністю відрізняється від вихідного й для добування інформаційного слова потрібно виконати операцію декодування й уже потім можна перевірити дані на зміст помилок. Таке кодування вимагає більші витрати ресурсів тільки на добування інформаційних даних, при цьому вони можуть бути без помилок.

При систематичному кодуванні до інформаційного блоку з k символів приписуються $2t$ перевірочних символів, при обчисленні кожного перевірочного символу використовуються всі k символів вихідного блоку.

У цьому випадку немає витрат ресурсів при добуванні вихідного блоку, якщо інформаційне слово не містить помилок, але кодер/декодер повинен виконати $k(n - k)$ операцій додавання й множення для генерації перевірочних символів. Крім того, тому що всі операції проводяться в поле Галуа, те самі операції кодування/декодування вимагають багато ресурсів і часу. Швидкий алгоритм декодування, заснований на швидкому перетворенні Фур'є, виконується за час порядку $l \ln n^2$.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Кодування

При операції кодування інформаційний поліном множиться на багаточлен, що породжує. Множення вихідного слова S довжини k на не приводиться полином, що, при систематичному кодуванні можна виконати в такий спосіб:

– До вихідного слова приписуються $2t$ нулів, виходить поліном $T = Sx^{2t}$.

– Цей поліном ділиться на поліном, що породжує, G , перебуває залишок R , $Sx^{2t} = QG + R$, де Q – частка.

– Цей залишок й буде коригувальним кодом Ріда-Соломона, він приписується до вихідного блоку символів. Отримане кодове слово $C = Sx^{2t} + R$.

Кодер будується зі регістрів зсуву, суматорів і перемножувачів. Регістр зсуву складається з комірок пам'яті, у кожній з яких перебуває один елемент поля Галуа.

Наведений як приклад кодер Ріда-Соломона генерує 16 коригувальних байт, що дозволяє виправляти до 8 і виявляти до 16 помилок у кадрі даних.

Перемножувачі на константи $GF(0) \dots GF(15)$ у полі Галуа реалізуються в такий спосіб: спочатку вихідне число й константа перетворюються в індексну форму, потім складаються в межах байта без обліку переносу.

Результатом операції є результат додавання, перетворена обернено в поліноміальну форму.

При переході від однієї форми подання даних до інший доцільно використовувати таблицю істинності розміром 256 байт, що становить ємність одного ЕАВ (Embedded Array Block – блок зосередженої пам'яті). Для реалізації кодеру потрібно 16 таких перемножувачів, при цьому те саме число множиться на різні константи, що дозволяє використовувати для його перекладу в індексну форму один ЕАВ.

Для перекладу результатів у поліноміальну форму потрібно вже 16 таких таблиць, що вимагає застосування ІМС FPGA дуже великої ємності. У запропонованій схемі використовується тактування кодера із частотою, в 8 разів перевищуючу частоту надходження байт даних.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Це дає можливість використовувати дві пари «суматор – ЕАВ», мультиплексує константи на входах суматорів і дозволяючи роботу регістрів-накопичувачів у моменти появи відповідних даних на виходах засувки ЕАВ.

Символу «С» відповідають дві константи $GF(n)$.

Символ «L» у логіці регістрів-накопичувачів відповідає наступний: вихід компаратора нуля (символи CMP0 і SYNC) дозволяє роботу схеми «АБО що виключає », на входи якої подаються вихід попереднього регістра й ЕАВ. Якщо ж вектор зворотного зв'язку дорівнює "0", схема пропускає дані з виходу попереднього регістра-накопичувача на вхід наступного.

У результаті кодер з урахуванням схеми синхронізації займає 255 LE (Logic Element – логічний елемент) і 3 ЕАВ, що дозволяє розмістити його в ІМС EPF10K10. Після оптимізації розміщення схеми на кристалі FPGA швидкодія схеми досягла 11,57 МГц (частота надходження байт даних, далі – байтова частота).

При використанні ІМС EPF10K20, у складі якої 6 ЕАВ, використовуючи 4 пари "суматор – ЕАВ", можна тактувати кодер із частотами, що перевищують байтову частоту не в 8, а в 4 рази, що дозволить підняти її до 25...30 МГц.

Декодування

Декодер, що працює по авторегресивному спектральному методі декодування, послідовно виконує наступні дії:

- Обчислює синдром помилки.
- Будує поліном помилки.
- Знаходить корінь даного полінома.
- Визначає характер помилки.
- Виправляє помилки.

Обчислення синдрому помилки

Обчислення синдрому помилки виконується синдромним декодером, що ділить кодове слово на багаточлен, що породжує. Якщо при діленні виникає остача, то в слові є помилка. Остача від ділення є синдромом помилки.

Знаходження кореня

На цьому етапі шукаються коріння полінома помилки, що визначають положення перекручених символів у кодовому слові. Реалізується за допомогою процедури Ченя, рівносильній повному перебору. У поліном помилок послідовно підставляються всі можливі значення, коли поліном звертається в нуль – коріння знайдені.

Визначення характеру помилки і її виправлення

По синдрому помилки й знайдених корінь полінома за допомогою алгоритму Форни визначається характер помилки й будується маска перекручених символів. Ця маска накладається на кодове слово за допомогою операції XOR і перекручені символи відновлюються. Після цього відкидаються перевірочні символи й виходить відновлене інформаційне слово.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерської роботи, наведена на рисунку 3.3.

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

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.

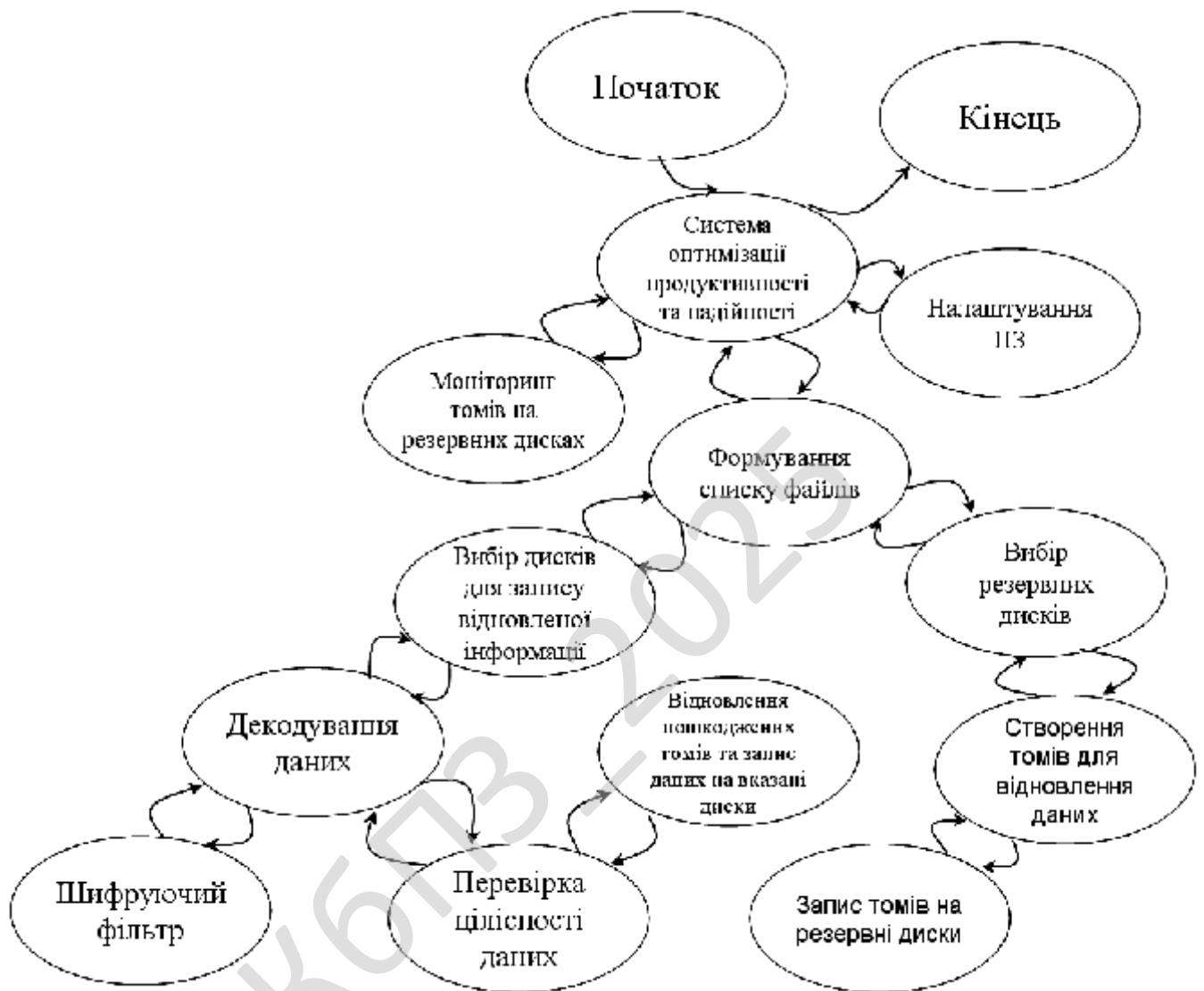


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи оптимізації продуктивності та надійності мережних систем зберігання.

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 підпрограми 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.

Під час роботи над магістерською роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

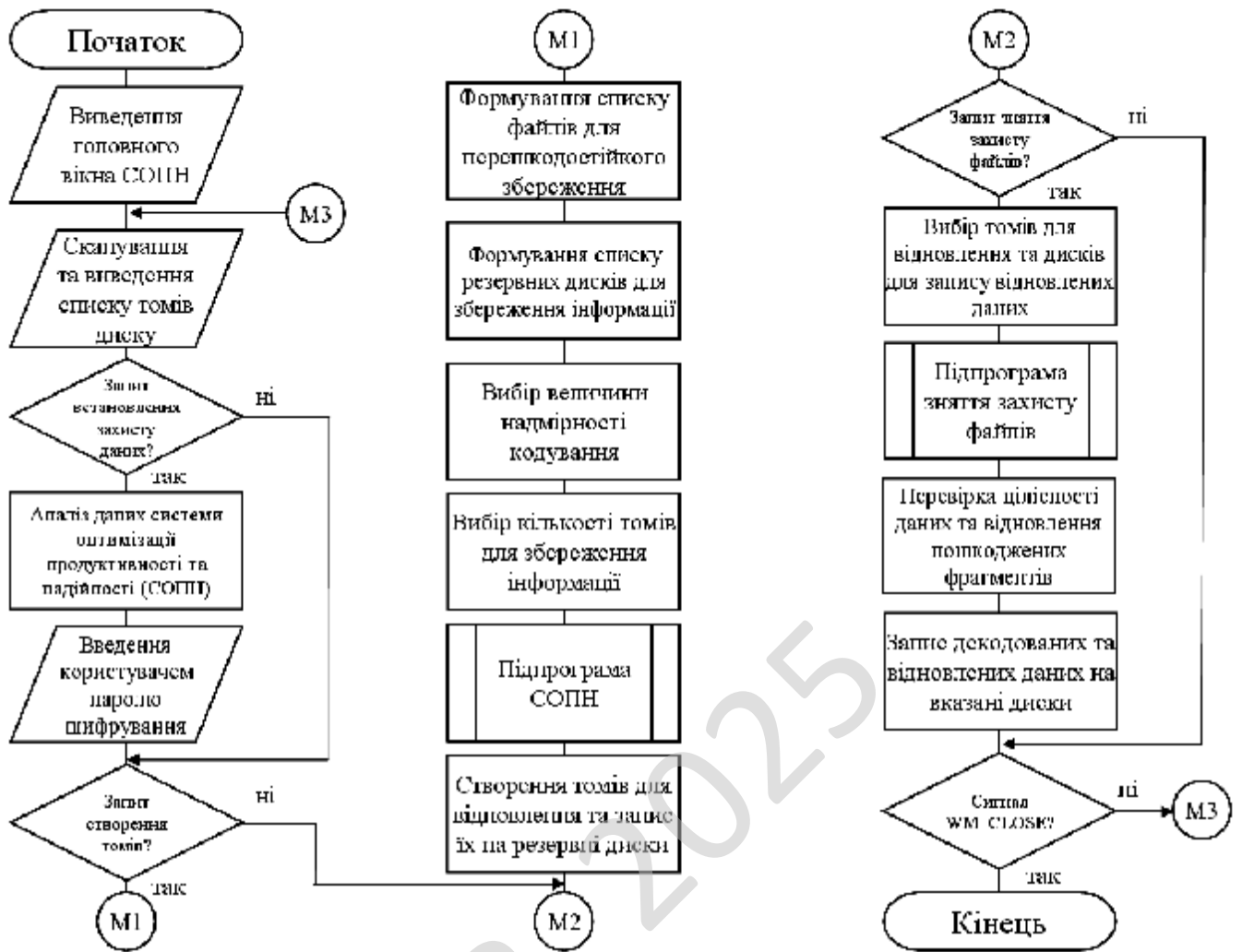


Рисунок 4.1 – Блок-схема основної програми

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Також при розробці магістерської роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів.

						ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			50

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

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

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

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

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

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

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

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

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

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

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

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

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

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

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

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

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

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

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

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

діапазон: "нуль або одиниця" (0..1), "багато" (0 .. *), "одиниця або більше" (1 .. *).
Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. *, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

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

Опис алгоритмів функціонування системи

Коди Ріда-Соломона (Reed-Solomon codes) – недвійкові циклічні коди, що дозволяють виправляти помилки в блоках даних. Елементами кодового вектора є не біти, а групи бітів (блоки). Дуже поширені коди Ріда-Соломона, що працюють з байтами (октет).

Код Ріда-Соломона є окремим випадком БЧХ-коду. Коди Боуза-Чоудхурі-Хоквінгема (БЧХ-коди) – в теорії кодування це широкий клас циклічних кодів, що

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

застосовуються для захисту інформації від помилок (попередня корекція помилок).

Відрізняється можливістю побудови коду із заздалегідь визначеними коригувальними властивостями, а саме, мінімальною кодовою відстанню. Окремим випадком БЧХ-кодів є Код Ріда-Соломона. Код винайшов в 1959 році А.Хоквінгом (Hocquenghem), і незалежно в 1960 році Р.Боуз (Bose) і Д.Рой-Чоудхурі (Ray-Chaudhuri). Код отримав свою назву (BCH code) від прізвищ їх авторів.

Коди БЧХ є узагальненням кодів Хеммінга і дозволяють виправляти кратні помилки.

В даний час широко використовується в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, в завадостійкому кодуванні.

Код Ріда-Соломона використовується при запису і зчитуванні в контролерах оперативної пам'яті, при архівуванні даних, запису інформації на жорсткі диски (ЕСС-пам'ять), запису на CD/DVD диски.

Навіть якщо пошкоджено значний обсяг інформації, зіпсовано кілька секторів дискового носія, то коди Ріда-Соломона дозволяють відновити велику частину втраченої інформації.

Також використовується при запису на такі носії, як магнітні стрічки і штрихкоди. Цей алгоритм кодування використовується при передачі даних по мережах WiMAX, в оптичних лініях зв'язку, у супутниковому та радіорелейному зв'язку.

Метод прямої корекції помилок в трафіку (Forward Error Correction, FEC) ґрунтується на кодах Ріда-Соломона.

Кодуємі дані передаються через масив $data[i]$, де $i = 0 \dots (k - 1)$, а згенеровані символи парності заносяться в масив $b[0] \dots b[2 * t - 1]$.

Вихідні й результуючі дані повинні бути представлені в поліноміальній формі (тобто у звичайній формі машинного подання даних).

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55


```

// на виході регістра буде частка, що губиться,
// а в самому регістрі - шуканий залишок
        for (j = n - k - 1; j > 0; j-- ) b[j] = b[j - 1]; b[0] = 0;
    }
}

```

Нижче наведемо вихідний текст підпрограми декодера Ріда-Соломона. Процедура декодування кодів Ріда-Соломона складається з декількох кроків. Спочатку ми обчислюємо $2t$ -символьний синдром шляхом постановки α^{**i} в $\text{rescd}(x)$, де rescd – отримане кодове слово, попередньо переведене в індексну форму. По факті обчислення $\text{rescd}(x)$ ми записуємо черговий символ синдрому в $s[i]$, де i приймає значення від 1 до $2t$, залишаючи $s[0]$ рівним нулю.

Потім, використовуючи ітеративний алгоритм Берлекэмп (Berlekamp), ми знаходимо поліном локатора помилки – $\text{elp}[i]$.

Якщо ступінь elp перевищує собою величину t , ми неспроможні скорегувати всі помилки й обмежуємося виводом повідомлення про непереборну помилку, після чого робимо аварійний вихід з декодера. Якщо ж ступінь elp не перевищує t , ми підставляємо α^{**i} , де $i = 1 \dots n$ в elp для обчислення корінь полінома. Обіг знайдений корінь дає нам позиції перекручених символів.

Якщо кількість певних позицій перекручених символів менше ступеня elp , перекручуванню піддалося більш ніж t символів і ми не можемо відновити їх.

У всіх інших випадках відновлення оригінального вмісту перекручених символів цілком можливо.

У випадку, коли кількість помилок свідомо велико, для їхнього виправлення декодуємо символи проходять крізь декодер без яких або змін.

```

decode_rs()
{
    int i, j, u, q;
    int s[n - k + 1];
    // поліном синдрому помилки
    int elp[n - k + 2][n - k];
    // поліном локатора помилки лямда
    int d[n - k + 2];
    int l[n - k + 2];

```

```

    int u_lu[n - k + 2],
int count = 0, syn_error = 0, root[t], loc[t], z[t + 1], err[n], reg[t + 1];
// переводимо отримане кодове слово в індексну форму
// для спрощення обчислень
    for (i = 0; i < n; i++) recd[i] = index_of[recd[i]];
// обчислюємо синдром
//-----
    for (i = 1; i <= n - k; i++)
    {
        s[i] = 0;          // ініціалізація s-регістра
                           // на його вхід за замовчуванням надходить нуль
// виконуємо s[i] += recd[j] * ij
// тобто беремо черговий символ декодуємих даних,
// множимо його на порядковий номер даного символу,
// помножений на номер чергового оберту й складаємо
// отриманий результат із умістом s-регістра;
// по факті вичерпання всіх декодуємих символ ми
// повторюємо весь цикл обчислень знову - по одному
// разу для кожного символу парності
for(j=0; j < n; j++) if (recd[j] != -1) s[i] ^= alpha_to[(recd[j] + i * j) % n];
        if (s[i] != 0) syn_error = 1;
// якщо синдром не дорівнює нулю, зводимо прапор помилки
// перетворимо синдром з поліноміальної форми в індексну
        s[i] = index_of[s[i]];
    }
// корекція помилок
// якщо є помилки, намагаємося їх скорегувати
    if (syn_error)
    {
        // обчислення полінома локатора ламбла
// обчислюємо поліном локатора помилки через ітеративний алгоритм
// Берлекемпа. Впливаючи термінологію Lin and Costello (див. "Error
// Control Coding: Fundamentals and Applications" Prentice Hall 1983
// ISBN 013283796) d[u] являє собою m ("мю"), що виражає
// розбіжність (discrepancy), де u = m + 1 і m є номер кроку
// з діапазону від -1 до 2t. У Блейхута та ж сама величина
// позначається D(x) ("дельта") і називається нев'язання.
// l[u] являє собою ступінь elp для даного кроку ітерації,
// u_l[u] являє собою різницю між номером кроку й ступенем elp

// ініціалізація елементи таблиці
        d[0] = 0;

```

						ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			58

Доповнення – вставка додаткових біт у кінець повідомлення для отримання кратності довжини бітової послідовності довжині внутрішнього стану функції гешування.

Повідомлення – бітова послідовність довжини від 0 біт (порожній рядок) до $2^{96}-1$ біт.

Функція стиснення – ітеративне перетворення, що відображає l -бітний блок повідомлення та l -бітне значення, отримане функцією стиснення на попередньому кроці, у нове l -бітне значення.

Далі використовуються наступні позначення:

- \oplus – додавання за модулем 2 (XOR);
- $0x$ – префікс числа, що записане у шістнадцятковій системі числення;
- $a \bmod b$ – ціле невід’ємне число, що дорівнює залишку від ділення цілого числа a на натуральне число b ;
- B_i – i -й байт вхідної послідовності;
- C^i – константа перетворення XORRoundKey або Add64RoundKey для i -го циклу;
- c – кількість стовпців внутрішнього стану в матричному поданні;
- ϕ – функція стиснення;
- H – визначена у стандарті функція гешування;
- $H(M)$ – результат обчислення функції гешування для повідомлення M (гешзначення);
- IV – вектор ініціалізації;
- l – розмір внутрішнього стану функції гешування (у бітах), $l \in \{512, 1024\}$;
- M – повідомлення;
- m_i – i -й блок повідомлення M ;
- n – довжина обчисленого геш-значення;
- N – довжина повідомлення M без доповнення;
- P, Q – складові перетворення функції стиснення;

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

- P_{512} – перетворення P для 512-бітного внутрішнього стану;
- P_{1024} – перетворення P для 1024-бітного внутрішнього стану;
- Q_{512} – перетворення Q для 512-бітного внутрішнього стану;
- Q_{1024} – перетворення Q для 1024-бітного внутрішнього стану;
- r – кількість ітерацій у перетвореннях P і Q ($r = t$ в ДСТУ 7564:2014);
- S – внутрішній стан геш-функції;
- t – кількість блоків m , з яких складається повідомлення M , включаючи

доповнення;

- v_i – i -й біт вхідної послідовності;
- Ω – завершальне перетворення;
- Купина- n – режим використання функції гешування з усіченням обчисленого гешзначення до розміру n біт.

Загальні положення

Під функцією гешування H розуміється залежне від вектора ініціалізації IV відображення послідовності біт M у геш-значення $H(M)$ фіксованої довжини n .

ДСТУ 7564:2014 визначає функцію гешування, яка виконує перетворення «Купина-256» або «Купина-512», що забезпечують обчислення геш-значення з довжинами 256 або 512 біт відповідно.

Геш-значення довжиною 256 бітів додатково може бути усічено до бітової послідовності довжиною від 8 до 248 біт з кроком у 8 біт, 512 бітів може бути усічене до бітової послідовності довжиною від 264 до 504 біт з кроком у 8 біт.

Режим роботи для формування геш-значення довжиною n біт позначається як «Купина- n ».

Основними режимами роботи функції гешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

Структура перетворення

Функція гешування, визначена в ДСТУ 7564:2014, формує геш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

(порожній рядок) до $2^{96}-1$ біт. При формуванні геш-значення повідомлення доповнюється, далі поділяється на l -бітні блоки m_0, \dots, m_t , після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення φ .

При цьому формуються значення $h_i = \varphi(h_{i-1}, m_i)$ де $i = 1, \dots, t$, а початкове значення $h_0 = IV$.

Після обробки останнього блоку повідомлення результуюче геш-значення обчислюється як $H(M) = \Omega(h_t)$, де Ω – завершальне перетворення, що повертає n - бітне значення, кратне 8 ($0 < n \leq l/2$).

КБПЗ_2025

					VKPM-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської роботи.

Розроблене програмне забезпечення системи оптимізації продуктивності та надійності мережних систем зберігання складається з наступних функціональних блоків:

– Навігаційне меню: Додатки; Налаштування системи оптимізації; Фільтри; Довідка.

– Розділи: Файли; резервні копії даних; Дії.

– Вікна обрання групи та функціональні кнопки ПЗ.

– Вікно виведення результату роботи системи.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

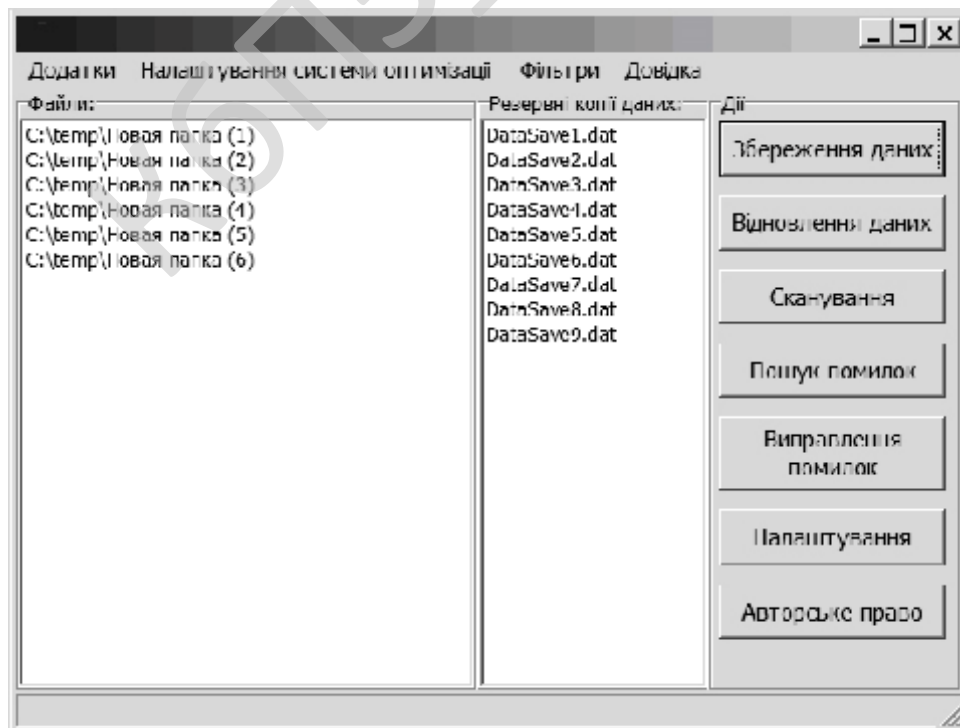


Рисунок 5.1 – Головне вікно розробленого ПЗ

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

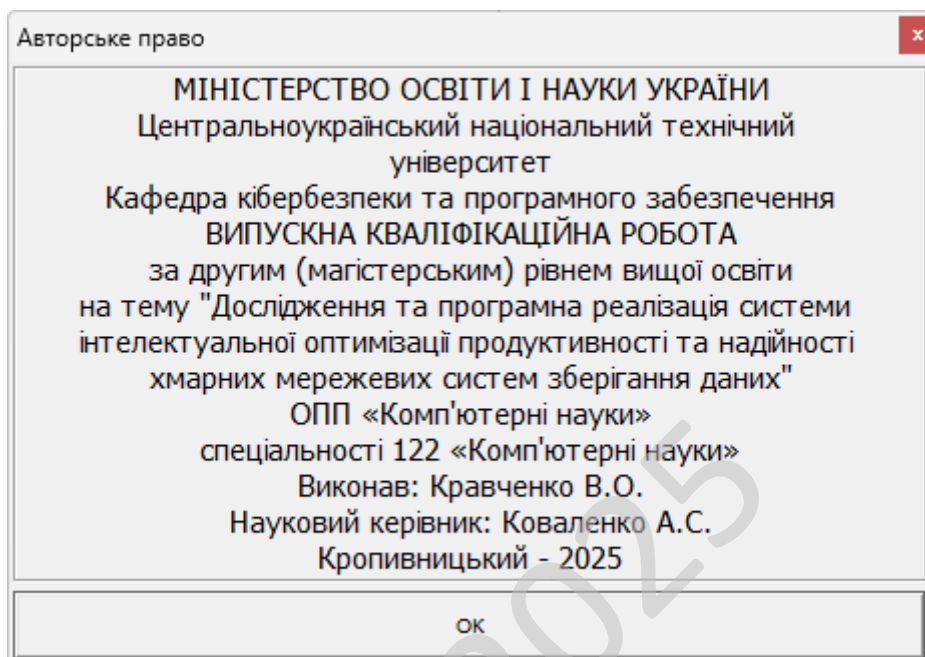


Рисунок 5.2 – Вікно розробника ПЗ

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

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

Проводилась оцінка:

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

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

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

Проводилось тестування форматом білої скриньки та чорної скриньки.

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

У цьому випадку формуються тестові варіанти, в яких:

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

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

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

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

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

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

Проводилось тестування чорної скриньки.

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

Ці тести демонструють:

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

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

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

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

Програмний виріб тут розглядається як «чорна скринька», 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).

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості. Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruватися), заплативши авторіві певну суму. В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КБПЗ_2025

					VKPM-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Метою розробки є дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Об'єктом дослідження є процес інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Предметом дослідження є методи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Методи дослідження базуються на методах теорії великих даних, теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

– Розроблено вітчизняний продукт інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати цього дослідження можуть бути особливо корисними для підприємств, що активно використовують хмарні сервіси у своїй діяльності. Для великих корпорацій і дата-центрів важливо не лише мати надійне сховище, але й оптимізувати витрати на його підтримку, забезпечити стабільну роботу при пікових навантаженнях і знизити ризики простоїв. Система інтелектуальної оптимізації дозволяє досягти цих цілей завдяки використанню алгоритмів машинного навчання, які передбачають потенційні проблеми ще до їх виникнення.

Також цей проєкт становить інтерес для постачальників хмарних послуг (Cloud Service Providers), адже впровадження таких технологій може стати вагомою конкурентною перевагою. Покращення швидкодії систем зберігання даних без масштабного оновлення обладнання дає змогу компаніям підвищувати ефективність бізнесу без значних капітальних витрат.

Крім того, розробка може бути цікавою для науково-дослідних установ і університетів, що займаються питаннями хмарних обчислень, штучного інтелекту та інформаційної безпеки. Вона може слугувати базою для подальших наукових розробок і впровадження нових концепцій управління ІТ-ресурсами. Також державні установи, які зберігають великі обсяги критичних даних, можуть використати подібні рішення для підвищення стабільності своєї ІТ-інфраструктури.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості проєкту було б доцільно залучити групу експертів – представників хмарних провайдерів, IT-директорів великих компаній і науковців у сфері оптимізації IT-інфраструктури. Їм пропонується оцінити систему за кількома критеріями: рівень інноваційності, потенціал ринку, технічна складність реалізації, очікувана економічна вигода та рівень ризику впровадження. Кожен експерт виставляє оцінку від 1 до 10 балів, після чого розраховується середньозважений показник привабливості.

Як приклад, якщо інноваційність оцінено на 9 балів, потенціал ринку – на 8, економічну вигоду – на 9, технічну складність – на 6 (через необхідність адаптації до існуючих систем), а рівень ризику – на 3, то середній індекс привабливості становитиме близько 8,2 балів із 10. Це свідчить про високий комерційний потенціал та актуальність продукту для ринку.

Такий метод дозволяє не лише оцінити перспективи розробки, а й визначити напрями для покращення – наприклад, підвищення зручності інтеграції або зменшення витрат на обслуговування. Отримані результати можуть бути використані для формування інвестиційної стратегії та залучення потенційних партнерів.

7.3 Вибір методу оцінки вартості ПЗ

Найдоцільнішим у цьому випадку буде метод дисконтованих грошових потоків (DCF) у поєднанні з витратним методом. DCF дозволяє врахувати не лише початкові витрати на розробку та впровадження, але й майбутні грошові потоки, які генеруватиме система завдяки зменшенню витрат на підтримку, уникненню простоїв і збільшенню ефективності роботи інфраструктури.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Витратний підхід допомагає деталізувати всі етапи реалізації: розробку програмного забезпечення, навчання моделей машинного навчання, закупівлю додаткового обладнання (якщо потрібно) і витрати на інтеграцію. Цей метод важливий для формування базової оцінки інвестиційного бюджету.

Комбінація двох методів надає найбільш реалістичну картину – можна оцінити не тільки скільки коштуватиме система, але й коли вона окупиться, та який обсяг економічного ефекту забезпечить. Такий підхід особливо цінний для компаній, які планують масштабне впровадження ІТ-рішень у власні дата-центри або для продажу на ринку хмарних технологій.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Хмарні системи зберігання даних є ключовим елементом сучасної ІТ-інфраструктури, однак із часом вони стикаються з проблемами продуктивності – зниження швидкості обробки запитів, надмірне навантаження на ресурси, затримки при передачі даних та нерівномірне розподілення навантаження між вузлами.

Впровадження інтелектуальної системи оптимізації (AI-based Cloud Performance Optimizer) дозволяє автоматично аналізувати метрики продуктивності, передбачати потенційні вузькі місця та динамічно переналаштовувати ресурси у режимі реального часу.

Основна мета проєкту – підвищення швидкодії, зменшення кількості відмов та зниження експлуатаційних витрат за рахунок прогнозування навантажень і проактивного управління хмарною інфраструктурою. Вхідні дані зафіксовано в таблиці 7.1.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

збільшення ефективності – +15 %, що еквівалентно 1 200 000 грн/рік додаткового прибутку, сукупний річний економічний ефект – 6 600 000 грн/рік, чистий економічний ефект – 6 200 000 грн/рік, термін окупності (Payback Period) \approx 0,45 року (5–6 місяців), рентабельність інвестицій (ROI) – 221 %.

Додаткові нефінансові переваги: покращення SLA (Service Level Agreement) – стабільність доступу до даних понад 99,98 %, зниження ризику відмов системи – завдяки проактивному моніторингу й прогнозуванню несправностей, підвищення задоволеності клієнтів – швидший доступ до даних та стабільна робота сервісів, автоматизація адміністрування – зменшення людського фактора і кількості рутинних дій, покращення масштабованості – система самостійно розподіляє ресурси при пікових навантаженнях.

Таким чином, проєкт забезпечує довгострокову вигоду у вигляді підвищення ефективності використання обладнання, стабільності хмарної інфраструктури та зниження ризиків втрати даних. Це дозволяє підприємству не лише зменшити витрати, а й створити основу для подальшої цифрової трансформації, підвищивши рівень довіри клієнтів і конкурентоспроможність на ринку IT-послуг.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Першим кроком просування проєкту має стати створення MVP (мінімально життєздатного продукту), який можна продемонструвати потенційним клієнтам. Демоверсія системи повинна показувати реальні переваги – наприклад, скорочення часу відповіді серверів, зниження затримок та підвищення ефективності зберігання даних. Це допоможе викликати довіру у бізнес-партнерів і сформувати початкову клієнтську базу.

Другим етапом є співпраця з хмарними провайдерами, такими як AWS, Google Cloud, Azure або локальні дата-центри, для пілотного впровадження. Результати тестування можна використовувати як кейси для подальшого

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

просування. Важливо активно висвітлювати успішні приклади використання у профільних ЗМІ та на конференціях з IT-інфраструктури.

На фінальному етапі доцільно запуснути маркетингову кампанію, орієнтовану на IT-директорів, адміністраторів дата-центрів і підприємства, що переходять на хмарні рішення. Просування має робити акцент на економії ресурсів, стабільності та можливості масштабування без великих капітальних інвестицій.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для підвищення ефективності реалізації проєкту слід орієнтуватися на модель SaaS (Software-as-a-Service), яка дозволяє користувачам отримувати доступ до системи через вебінтерфейс без потреби встановлення локального програмного забезпечення. Це значно спрощує процес впровадження та скорочує бар'єри для входу на ринок.

Іншим напрямом може бути співпраця з інтеграторами IT-рішень, які вже мають клієнтів у сфері хмарних технологій. Такі партнери зможуть пропонувати систему як частину комплексного рішення для оптимізації інфраструктури. Також варто створити гнучку систему ліцензування – наприклад, на основі кількості вузлів або обсягу оброблених даних.

Додатково можна розробити навчальні програми та сертифікацію для IT-спеціалістів, які працюватимуть із системою. Це підвищить впізнаваність бренду й забезпечить стабільний попит серед технічних фахівців.

7.7 Визначення ключових факторів успіху конкретного проєкту

Основними факторами успіху цього проєкту є якість і точність роботи алгоритмів штучного інтелекту, на яких базується система. Якщо вона зможе ефективно прогнозувати навантаження, автоматично оптимізувати розподіл

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

ресурсів і попереджати відмови обладнання, то її цінність для клієнтів буде очевидною.

Не менш важливим чинником є простота інтеграції з існуючими хмарними платформами та можливість масштабування під потреби різних підприємств. Гнучкість системи та адаптивність до різних типів хмар – приватних, публічних чи гібридних – значно розширить її ринковий потенціал.

Також успіх залежить від ефективної комунікації з ринком: чітке позиціонування, прозора демонстрація вигод і зворотний зв'язок із користувачами. Якщо розробники зможуть поєднати технічну досконалість із практичною користю для бізнесу, система стане популярною не лише як інновація, а як реальний інструмент оптимізації витрат і підвищення ефективності IT-інфраструктури.

КБПЗ_2025

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Комп'ютерні мережі вриваються у життя людей в їх професійну діяльність найнесподіванішим і масовим чином. Вони породили істотно нові технології обробки інформації – мережні технології. У найпростішому разі мережні технології дозволяють спільно використовувати ресурси – нагромаджувачі великий ємності, друкують устрою, доступ в Internet, бази й банки даних. Найбільш сучасні і перспективні підходи до мереж пов'язані з допомогою колективного поділу праці – розробці різних документів і майже проектів, управлінні установою чи підприємством і т. д.

Впровадження комп'ютерних технологій принципово змінило характер праці різних категорій фахівців. Працівники, використовують комп'ютерну техніку, на своєму досвіді оцінили її величезні можливості. Одночасно виникла певна безтурботність при її експлуатації.

Недотримання вимог безпеки призводить до того, що й через кілька днів роботи за комп'ютером співробітник починає відчувати певний дискомфорт: в нього виникає головний біль і різь у власних очах, з'являються почуття виснаження й дратівливості. В окремих людей порушується сон, погіршується зір, занедужують руки, шия, поперек тощо.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;

- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці.

8.2 Аналіз умов праці

Приміщення розташовано на третьому поверсі п'ятиповерхового будинку. У приміщенні розташовано 3 робочих місця з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м², об'єм не менше 20 м³. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м². Необхідна площа на 3 робочих місця із установленими ПК складає 18 м², що не перевищує фактичну. Обсяг приміщення на одного працюючого складає 31,5 м³, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м³.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань. Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. Шкідливі і небезпечні чинники, з якими стикається ІТ-працівник при роботі з ПК, приведені в таблиці 8.1.

Таблиця 8.1 – Перелік шкідливих та небезпечних виробничих факторів

Найменування факторів	Можливі джерела їх виникнення	Характер дії
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що згорають і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Іонізація повітря	Статична електрика випромінювання	Шкідливий
Підвищений рівень шуму	Шум створюється перетворювачем напруги ЕОМ, її технічною периферією, а також людьми, що працюють в приміщенні	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Незадовільні параметри мікроклімату	Незадовільний стан системи опалення і вентиляції	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

По категорії вибухо- і пожежонебезпеки, згідно дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії пожежонебезпеки і поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [3] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

По ступені небезпеки поразки людей електричним струмом відділ, згідно, класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціонування в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціонування» [4]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам відповідно до ДСанПіНЗ.3.2.007 – 98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно – обчислювальних машин».

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м³.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [5]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

Одним з найбільш поширеніших чинників виробничого середовища, який несприятливо впливає на людину, є шум. Вплив шуму на організм людини

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Загальний час роботи – робочий день, тобто $T=8$ годин.

Для фонового шуму (вентиляторів):

$$L_1 = 35 \text{ дБА}, T_1 = 8 \text{ годин}, n_1=15 (5 \times 3);$$

Для лазерного принтера Lexmark Jet:

$L_2 = 48 \text{ дБА}, T_2 = 2 \text{ години}, n_2=1$, для сканера $L_3 = 46 \text{ дБА}, T_3 = 2 \text{ години}$.

Підставляємо отримані величини у формулу (8.1):

$$L_{\text{СКВ}} = 10 \cdot \lg \left(\frac{1}{8} \cdot (15 \cdot 8 \cdot 10^{0,1 \cdot 35} + 1 \cdot 2 \cdot 10^{0,1 \cdot 48} + 1 \cdot 2 \cdot 10^{0,1 \cdot 46}) \right) = 46,3 \text{ дБА}$$

Таким чином, еквівалентний рівень шуму в приміщенні за робочий день $L_{\text{екв}} = 46,3 \text{ дБА}$, тобто не перевищує норму 50 дБА .

8.3 Техніка безпеки та протипожежна профілактика

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізолюваний кабель. Висота проводки становить $2,2 \text{ м}$ від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1 м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення.

Всі обчислювальні машини з'єднані із клемою заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V , а одна, забезпечує подачу напруги в 36 V . Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежобезпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки.

До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електроні справності встаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрана дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту.

Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропонується використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості, проведених відділом охорони праці.

В якості мір по зниженню шуму можна запропонувати:

- облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 до);
- екранування робочого місця (встановленням перегородок, діафрагм);

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

- установка в комп'ютерних приміщеннях устаткування, що викликає мінімальний шум;
- раціональне планування приміщення.

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

8.5 Висновки до розділу

У даному розділі магістерської роботи були виконано аналіз умов праці користувачів ПК, які працюють у зазначеному приміщенні. Проведено перевірку організації робочого місця із відповідними замірами параметрів мікроклімату, освітлення, рівня шуму та розрахунком рівня шуму.

Розроблені заходи щодо поліпшення умов праці дотримання техніки безпеки та проведення протипожежної профілактики дозволить створити умови, які будуть забезпечувати більш комфортну роботу.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.
- Досліджена система інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.
- На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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 Visual C#. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7564:2014.

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.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кравченко В.О. Дослідження та програмна реалізація системи інтелектуальної оптимізації продуктивності та надійності хмарних мережевих систем зберігання даних // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.

3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.

4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.

5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.

6. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.

7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.

8. 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.

9. 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.

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

10. 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.

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

12. 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.

13. 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.

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

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

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

24. 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.

25. 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.

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

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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. .

44. 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.

45. Смірнов О.А., Дрєєва Г.М., Дрєєв 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. .

					ВКРМ-122.25.0043.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

46. 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.

47. 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 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).

48. 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.

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

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

51. 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.