

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 дедуплікації в
інфраструктурі резервного копіювання”**

КБПЗ - 2025

Виконав здобувач вищої освіти
II курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Кривда О.С.
« ____ » _____ 2025 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
« ____ » _____ 2025 р.
Рецензент _____

АНОТАЦІЯ

Кривда О.С. Дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи дедуплікації в інфраструктурі резервного копіювання.

Метою розробки є дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання.

Об'єктом дослідження є процес дедуплікації в інфраструктурі резервного копіювання.

Предметом дослідження є методи дедуплікації в інфраструктурі резервного копіювання.

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

Результат роботи – програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання.

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'ютерна інженерія, дедуплікація, резервне копіювання

ABSTRACT

Kryvda O.S. Research and software implementation of a deduplication system in a backup infrastructure. 123 Computer Engineering. 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 a deduplication system in a backup infrastructure.

The purpose of the development is the research and software implementation of a deduplication system in a backup infrastructure.

The object of the research is the deduplication process in a backup infrastructure.

The subject of the research is deduplication methods in a backup infrastructure.

The research methods are based on the methods of the theory of building computer networks, methods of mathematical statistics, and methods of software development.

The result of the work is a software implementation of a deduplication system in a backup infrastructure.

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 user-friendly 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 engineering, deduplication, backup

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.....	10
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	16
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	16
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	37
2.3 Розгорнута постановка завдання	41
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	42
3.1 Опис функціонування системи	42
3.2 Розробка структурної схеми.....	47
3.3 Розробка функціональної схеми	51
3.4 Розробка діаграми процесів.....	67
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	70
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	70
4.2 Захист розробленого програмного забезпечення.....	84
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	89
6 НАУКОВА НОВИЗНА	96

					ВКРМ-123.25.0046.00.00.ПЗ			
Вим	Арк	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання	Літ.	Аркуш	Аркушів
Розроб.	Кривда О.С.					М	1	120
Перев.	Якименко Н.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-24М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	97
7.1	Визначення цільової аудиторії кінцевого готового продукту	97
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	98
7.3	Вибір методу оцінки вартості ПЗ	98
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	99
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	101
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	101
7.7	Визначення ключових факторів успіху конкретного проєкту.....	102
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	103
8.1	Вступ.....	103
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	104
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста .	106
8.4	Розробка заходів з умов поліпшення охорони праці	109
8.5	Розрахункова частина	110
9	ОСНОВНІ ВИСНОВКИ.....	112
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	114

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
BIOS	–	Basic Input-Output System
LINQ	–	Language Integrated Query
RAID	–	Redundant Array of Inexpensive Disks

КБПЗ – 2025

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Дедуплікація знайшла широке застосування як спосіб упоратися з ростом витрат на інфраструктуру резервного копіювання при збільшенні обсягів даних.

Доступність і схоронність даних – застава безперервності бізнес-процесів і ефективності роботи. Сьогодні дані – один із самих коштовних активів, тому їхнє резервне копіювання й архівне зберігання – найбільш типові завдання, а система резервного копіювання – важлива частина будь-якої корпоративної інформаційної системи. При правильній організації вона здатна надійно захистити критичні дані.

Впровадження систем резервного копіювання дає можливість оперативно відновлювати інформацію в самих різних ситуаціях, однак і вони не позбавлені недоліків. Традиційні проблеми – неефективне використання ємності зберігання (обсяг резервних копій за тиждень може вдвічі перевищити обсяг вихідних даних), низька швидкість копіювання, непередбачений час відновлення (як правило, набагато більше планованого), не дуже висока надійність (по даним Gartner, ризик неможливості відновлення перевищує 10%).

Рішення для резервного копіювання й відновлення повинні ефективно функціонувати в умовах експонентного росту даних, жорсткості вимог регуляторів і скорочення вікон резервного копіювання. Українцям бажано також забезпечити зниження витрат, пов'язаних із захистом даних. Перебороти перераховані проблеми покликана дедуплікація даних. У том або іншому виді її пропонують у своїх продуктах всі провідні вендори систем резервного копіювання корпоративного класу.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

- Огляд існуючих систем дедуплікації в інфраструктурі резервного копіювання.
- Дослідження системи дедуплікації в інфраструктурі резервного копіювання.
- Програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання.

Об'єктом дослідження є процес дедуплікації в інфраструктурі резервного копіювання.

Предметом дослідження є методи дедуплікації в інфраструктурі резервного копіювання.

Методи дослідження базуються на методах 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 дедуплікації в інфраструктурі резервного копіювання.
- Розроблено вітчизняний продукт дедуплікації в інфраструктурі резервного копіювання, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі дедуплікації в інфраструктурі резервного копіювання.

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-123.25.0046.00.00.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 дедуплікації в інфраструктурі резервного копіювання, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2025

					VKPM-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Дедуплікацію даних для стислості часто називають Dedup. Це функція, за допомогою якої можливо зменшити вплив надлишкових даних на вартість зберігання. Якщо дедуплікація даних включена, вона оптимізує вільне місце в томуї за рахунок перевірки дані томи на наявність частин, що дублюються. частини, Що Дублюються, набору дані томи зберігаються один раз і (при необхідності) стискаються для додаткової економії. Дедуплікація оптимізує надлишкові дані, не порушуючи вірогідність або цілісність даних.

Переваги дедуплікації даних

Дедуплікація даних допомагає адміністраторам сховища знизити витрати, пов'язані з даними, що дублюються. Найчастіше в великих наборах даних багато даних дублюються, що збільшує витрати на їхнє зберігання. Наприклад:

- Файлові ресурси користувачів можуть містити безліч копій тих самих або схожих файлів.
- Гостьові служби віртуалізації можуть практично не відрізнятися від служб на віртуальних машинах.
- Моментальні знімки резервних копій можуть мати мінімальні відмінності від щоденних.

Економія місця, що може забезпечити дедуплікація даних, залежить від набору даних або робочого навантаження в томуї. У наборах даних з високим рівнем дуплікації швидкість оптимізації досягає 95%, а обсяг використання служби сховища може зменшуватися в 20разів. У наступній таблиці представлені типові значення економії за рахунок дедуплікації для різних типів умісту.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Таблиця 1.1 – Типові значення економії за рахунок дедуплікації для різних типів умісту

Сценарій	Уміст	Звичайна економія простору
Документи користувача	Документи Office, фотографії, музика, відео й т.п.	30-50 %
Загальні ресурси розгортання	Двійкові файли програмного забезпечення, САВ-Файли, символи й т.д.	70–80 %
Бібліотеки віртуалізації	Образи ISO, файли віртуальних жорстких дисків і т.д.	80–95 %
Файловий ресурс загального доступу	Все перераховане вище	50–60 %

Коли можливо використовувати дедуплікацію даних

Файлові сервери загального призначення

Файлові сервери загального призначення являють собою файлові сервери для загального використання, які можуть містити загальні папки будь-якого типу з перерахованих далі:

- загальні групові папки;
- домашні папки користувачів;
- робочі папки;
- загальні ресурси для розробки програмного забезпечення.

Файлові сервери загального призначення підходять для дедуплікації даних через тенденцію збереження численних копій або версій одного файлу декількома користувачами. Від дедуплікації даних виграють загальні ресурси для розробки програмного забезпечення, тому що багато двійкових файлів залишаються по суті незмінними від складання до складання.

Розгортання інфраструктури віртуальних робочих столів (VDI)

Завдяки серверам VDI, таким як служби віддалених робочих столів, організації одержують спрощений спосіб підготовки настільних комп'ютерів для користувачів. Ця технологія підходить для організацій з багатьох причин:

– **Розгортання застосунків:** ви одержуєте можливість швидкого розгортання застосунків у середовищі підприємства. Це особливо корисно при наявності застосунків, які часто обновляються, рідко використовуються або є складними в керуванні.

– **Консолідація застосунків:** при установці й запуску застосунків з набору централізовано керованих віртуальних машин більше немає необхідності обновляти застосунку на клієнтських комп'ютерах. Це також знижує вимоги до пропускної здатності мережі, необхідної для доступу до застосунків.

– **Віддалений доступ:** користувачі можуть одержувати доступ до корпоративних застосунків з таких пристроїв, як домашні комп'ютери, кіоски й малопотужне встаткування, а також з операційних систем, відмінних від Windows.

– **Доступ до філій:** розгортання VDI можуть забезпечити кращу продуктивність застосунків для працівників філії, яким потрібен доступ до централізованих сховищ даних. Ресурсомісткі додатки іноді не мають протоколів клієнтів і серверів, оптимізованих для підключень по повільній лінії.

Розгортання VDI прекрасно підходять для дедуплікації даних, тому що віртуальні жорсткі диски, що визначають віддалені робітники столи для користувачів, по суті ідентичні. Крім того, дедуплікація даних може допомогти у випадку падіння продуктивності сховища на пікових навантаженнях (так званий VDI boot storm), коли безліч користувачів одночасно входить на настільні системи на початку дня.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Цільові об'єкти резервного копіювання, наприклад віртуалізовані застосунку резервного копіювання

Застосунку резервного копіювання, такі як Microsoft Data Protection Manager (DPM), прекрасно підходять для дедуплікації даних, тому що значна частина моментальних знімків резервних копій дублюється.

Інші робочі навантаження

До інших робочих навантажень також можливо застосовувати дедуплікацію даних.

1.2 Область застосування

Дедуплікація даних розроблялася на основі двох найважливіших принципів.

1. Оптимізація не повинна заважати процесам запису на диск. Дедуплікація даних оптимізує дані за допомогою моделі постобробки. Всі дані записуються на диск у неоптимізованому виді, а потім оптимізуються за допомогою дедуплікації даних.

2. Оптимізація не повинна змінювати семантику доступу до даних. Користувачі й додатки, звертаючись до даних в оптимізованому томі, не повинні навіть знати про те, що до цих даних застосована дедуплікація.

Після включення дедуплікації даних для тому вона виконує у фоновому режимі наступні завдання:

- виявляє повторювані фрагменти у файлах тому;
- автоматично переміщає ці фрагменти (блоки) зі спеціальними покажчиками, які називаються точками повторного аналізу й указують на унікальну копію блоку.

Цей процес виконується в п'ять етапів:

- перевірка файлової системи на наявність файлів, що відповідають політиці оптимізації;

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

- розбивка файлів на блоки змінного розміру;
- виявлення унікальних блоків;
- приміщення блоків у сховище блоків зі стиском, якщо застосовно;
- заміна вихідного потоку даних в оптимізованих файлах на точки повторного аналізу, що вказують на сховище блоків;

При зчитуванні оптимізованих файлів файлова система відправляє файли із точкою повторного аналізу у фільтр дедуплікації даних файлової системи (Dedup.sys). Фільтр перенаправляє операцію читання до відповідних блоків, які утворять потік цього файлу в сховище блоків. Зміни фрагментів дедуплікованого файлу записуються на диск у неоптимізованому виді. Їх при наступному запуску обробляє завдання оптимізації.

Типи використання

Наступні типи використання містять раціональні налаштування дедуплікації даних для деяких розповсюджених робочих навантажень.

Таблиця 1.2 – Типи використання містять раціональні налаштування дедуплікації даних

Тип використання	Підходящі робочі навантаження	Відмінності
Значення за замовчуванням	Файловий сервер загального призначення. – загальні групові папки – робочі папки – перенапрямок папок. – загальні ресурси для розробки програмного забезпечення	– фонові оптимізація – політика оптимізації за замовчуванням: – мінімальний вік файлу – 3 дні; – оптимізація використовуваних файлів – немає; – оптимізація неповних файлів – немає.

Оптимізація

Завдання оптимізації виконує дедуплікацію, розбиваючи на блоки дані, що зберігаються на томі, відповідно до налаштувань політики для цього тому, а також (необов'язково) стискаючи ці блоки й зберігаючи їхні унікальні копії в сховище блоків. Процес оптимізації, використовуваний дедуплікацією даних, докладно описаний у розділі Як працює дедуплікація даних?

Збір сміття

Завдання складання сміття виконує звільнення місця на диску, видаляючи непотрібними блоки, що стали, на які не залишилося посилань після зміни або видалення файлів.

Перевірка цілісності

Завдання перевірки цілісності виявляє ушкодження в сховище блоків, зв'язані зі збоями диска або ушкоджених секторів. У міру можливості дедуплікація даних автоматично застосовує доступні для тому функції (наприклад, дзеркала або контроль парності для тому дискових просторів), щоб відновити ушкоджені дані. Крім того, дедуплікація даних зберігає в окремій "активній зоні" резервні копії популярних блоків, на які існує більше 100 посилань.

Скасування оптимізації

Завдання скасування оптимізації, особливе завдання, що може виконуватися тільки вручну, скасовує всю оптимізацію, виконану службою дедуплікації, і відключає дедуплікацію даних для тому.

Глоссарий дедуплікації даних

Блок

Блоком називається фрагмент файлу, відібраний алгоритмом дедуплікації даних, що з високою часткою ймовірності буде повторюватися в інших схожих файлах.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Сховище блоків

Сховище блоків – це впорядкований набір файлів у папці "System Volume Information", що дедуплікація даних використовує винятково для зберігання блоків.

Dedup

Скорочена форма англomовної назви дедуплікації даних, що часто використовується в PowerShell, інтерфейсах API і компонентах Windows Server, а також у співтоваристві Windows Server.

Метадані файлу

Кожний файл містить метадані, які описують важливі властивості файлу, не зв'язані прямо з основним змістом файлу. Наприклад: дата створення файлу, дата останнього читання, творець файлу й т.д.

Файловий потік

Так називається основний зміст файлу. Саме цю частину файлу оптимізує дедуплікація даних.

Файлова система

Файловою системою називають спеціалізоване програмне забезпечення й структуру даних, що зберігаються на диску, які використовуються операційною системою для зберігання файлів на будь-яких носіях. Дедуплікація даних підтримується тільки на томах з файловою системою NTFS.

Фільтр файлової системи

Так називається підключається модуль, що, що змінює стандартне поведіння файлової системи. Щоб зберегти семантику доступу, дедуплікація даних використовує фільтр файлової системи (Dedup.sys), що перенаправляє запити на читання оптимізованого вмісту непомітним для користувача або застосунку образом.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Оптимізація

Файл вважається оптимізованим з погляду дедуплікації даних (дедуплікованим), якщо він розділений на унікальні блоки, які перенесені в сховище блоків.

Політика оптимізації

Політика оптимізації визначає, для яких файлів варто застосовувати дедуплікацію даних. Наприклад, політика може виключати з оптимізації недавно створені або відкриті файли, всі файли в певнім розташуванні в томуі або файли певного типу.

Точка повторного аналізу

Точкою повторного аналізу називають спеціальний тег, що повідомляє файлову систему про необхідність перенаправляти запит вводу-виводу на зазначений фільтр файлової системи. У тих файлах, для яких виконана оптимізація, дедуплікація даних заміняє файловий потік точкою повторного аналізу, що дозволяє повністю зберігати семантику доступу до цього файлу.

Том

Том – це використовуване Windows позначення для логічного диска зберігання даних, що може включати кілька фізичних пристроїв зберігання, розташованих на одному або декількох серверах. Дедуплікація включається на рівні окремого тому.

Робоче навантаження

Робочим навантаженням називається додаток, виконуване на Windows Server. Приклад робочого навантаження – файловий сервер загального призначення, сервер Hyper-V і SQL Server.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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

Розглянемо деякі системи резервного копіювання більш детально.

IBM Spectrum Protect і Protectier

Система IBM Spectrum Protect дозволяє зменшити ризик втрати даних за рахунок постійного інкрементного резервного копіювання й дедуплікації. Вона підтримує безліч різних видів сховищ, у тому числі гібридні хмари, і допомагає автоматизувати керування інформацією.

IBM утримує позицію одного із провідних виробників систем резервного копіювання, у тому числі завдяки запуску технології хмарного багаторівневого зберігання для IBM Spectrum Protect. Це рішення дозволяє здійснювати безпечно й простої в керуванні автоматичне резервне копіювання даних у хмарі.

IBM TS7650G ProtecTIER Deduplication Gateway володіє одними із кращих у галузі показниками швидкості дедуплікації в реальному часі. Ємність сховища резервних копій може перевищувати 25 Пбайт. У сполученні із системами зберігання даних (від IBM або інших вендорів) TS7650G ProtecTIER підвищує продуктивність сховища й забезпечує довгострокове зберігання й доступність даних, що перебувають у резервних копіях і архівах.

Продуктивність резервного копіювання при дедуплікації віртуальної стрічкової бібліотеки (VTL), здійснюваної в реальному часі, при передачі даних досягає 9 Тбайт у годину. За інформацією IBM, у системі в компактному корпусі 2U застосовується алгоритм підтримки цілісності даних без гешування, тому ризик втрати даних через колізії геш-функцій зводиться до нуля.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Завдяки патентованому алгоритму дедуплікації HyperFactor, що не використовує геш-функції, необхідна ємність системи зберігання даних зменшується в 25 разів і більше. На думку розроблювачів, з його допомогою можливо істотно знизити витрати й забезпечити корпоративний рівень цілісності даних. Індекс зберігається в оперативній пам'яті й не утрудняє обробку більших масивів даних (до 10 Тбайт), що не виключено у випадку гешування.

Додаток резервного копіювання записує дані на ProtecTIER як у звичайну стрічкову бібліотеку. При цьому зберігається тільки унікальна інформація, на вже існуючу створюються посилання. Коли дані застарівають, посилання видаляються й місце звільняється.

Avamar від Dell EMC

Дедуплікація сегментів змінної довжини, що здійснюється на пристрої клієнта, реалізована в системі резервного копіювання Avamar від Dell EMC. Avamar – комплексне програмно-апаратне рішення для резервного копіювання й відновлення даних, інтегроване зі СЗД Data Domain. Воно підтримує віртуальні й фізичні середовища, корпоративні додатки, мережні системи зберігання даних (NAS) і ПК, захист даних віддалених офісів.

Глобальна дедуплікація на стороні клієнта зменшує обсяг резервного копіювання. Специфіка глобальної дедуплікації полягає в тому, що при будь-якому типі або кількості зовнішніх пристроїв і сеансів дані копіюються й зберігаються в пулі дедуплікації один раз. Як відзначають в Dell EMC, скорочення часу на щоденне резервне копіювання досягає 90%, навантаження на мережу (у термінах необхідної пропускної здатності) зменшуються на 99%, а сумарна ємність дискових систем зберігання даних – на 95%.

Система розділяє даному, підлягаючому резервному копіюванню, на сегменти, стискає їх і застосовує для кожного унікальний геш-ідентифікатор. Потім вона визначає, чи вироблялося раніше резервне копіювання сегмента, і копіює тільки унікальну інформацію. Резервне копіювання тих самих даних ніколи не виконується.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Швидке одноетапне відновлення даних виключає необхідність відновлювати останні цілісні повні й інкрементні резервні копії. Надійність сервера Avamar і можливість відновлення даних резервного копіювання перевіряються щодня. Рішення дозволяє оперативно відновити дані конкретного застосунку.

Avamar можливо розгортати й у вигляді тільки програмного рішення. EMC Avamar Virtual Edition (AVE) являє собою віртуальний пристрій з функцією дедуплікації для резервного копіювання й відновлення. AVE дозволяє розгорнути повнофункціональний сервер Avamar у середовищі віртуалізації VMware або Microsoft Hyper-V. Коли як система зберігання використовується Data Domain Virtual Edition, AVE можливо масштабувати до 16 Тбайт.

Серед технологічних особливостей рішення можливо виділити трохи найбільш корисних: наявність спеціального кеша унікальних файлів і блоків, що дозволяє здійснювати обхід файлової системи значно швидше, ніж у випадку традиційного резервного копіювання; підтримку більшості відомих корпоративних застосунків, таких як SAP, Oracle, MS SQL і інших; підтримку технології VMware Changed Block Tracking (CBT) для відстеження змінених блоків (дозволяє прискорити процес резервного копіювання й відновлення), а також наявність спеціального плагіну для vCenter для керування прямо із цієї консолі.

Серед нових можливостей продукту – підтримка хмар і багаторівневе зберігання. Крім того, Avamar також може запропонувати варіант організації довгострокового зберігання в приватній хмарі – як з використанням системи Data Domain, так і без її. Data Domain Cloud Tier підтримує хмари Azure, Amazon, Virtustream і будь-які пристрої, що використовують протокол S3. Хмарне резервне копіювання – одна з тенденцій у резервуванні даних.

Дедуплікація даних в Windows Server 2025

Можливість дедуплікації даних уперше з'явилася в Windows Server 2025. З тих пор пройшло багато часу й в Windows Server 2025 включена вже третя версія

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

дедуплікації, перероблена й поліпшена. Про те, що саме вміє нова дедуплікація й чим вона відрізняється від попередніх реалізацій.

Перша й найбільш важлива зміна в роботі дедуплікації Windows Server 2025 – це введення багатопоточності.

Багатопоточність

В Windows Server 2025 R2 дедуплікація працює в однопотоковому режимі й може задіяти не більше 1 процесорного ядра на один том. Це серйозно обмежує продуктивність, а для обходу цього обмеження необхідно розбивати диски на кілька томів меншого розміру. При цьому максимальний розмір тому не повинен бути більше 10Тб.

В Windows Server 2025 двигок дедуплікації перероблений і завдання дедуплікації може працювати в багатопоточному режимі, використовуючи для кожного тому відразу кілька обчислювальних потоків і черг вводу\виводу.

Щоб переконатися в тому, що багатопоточність дійсно працює, я провів невеликий експеримент. Для експерименту я взяв сервер із двома 6-ядерними процесорами, що при включеному HyperThreading дає нам 24 логічних процесора. На сервері була встановлена Windows Server 2025 R2 і включена дедуплікація для одного тому.

Я запустив на сервері процедуру оптимізації, замірив продуктивність і з'ясував, що процес дедуплікації (fsdmhost.exe) використовує **максимум** 4% потужності процесора, що приблизно відповідає 1 логічному ядру.

Потім операційна система на сервері була проапгрейжена на Windows Server 2025, після чого я повторно замірив продуктивність. І як бачите, після апгрейда дедуплікація вже не обмежена одним ядром і може використовувати більше 4% процесорної потужності сервера.

Впровадження багатопоточності й інші зміни у двигку вплинули на обмеження по розмірі томів і файлів. Оскільки багатопоточність збільшує продуктивність дедуплікації й знімає необхідність у розбивці диска на кілька томів, в Windows Server 2025 для дедуплікації можливо використовувати томи до

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

64Тб. Також збільшений максимальний розмір файлу, тепер підтримується дедуплікація файлів розміром до 1Тб.

Підтримка віртуалізованих застосунків резервного копіювання

В Windows Server 2025 був усього один тип дедуплікації, призначений в основному для звичайних файлових серверів. Дедуплікація таких навантажень, що як постійно працюють віртуальні сервера, не підтримувалася, оскільки дедуплікація не вміла працювати з відкритими файлами.

В Windows Server 2025 R2 дедуплікація навчилася використовувати VSS, відповідно стала підтримуватися дедуплікація віртуальних серверів. Для таких навантажень з'явився окремий тип дедуплікації.

В Windows Server 2025 додався ще один, третій тип дедуплікації, призначений спеціально для віртуалізованих серверів резервного копіювання (напр. DPM). Треба сказати, що такий варіант був і раніше, однак у зв'язку зі специфічними навантаженнями налаштування дедуплікації було потрібно підбирати вручну. В Windows Server 2025 для таких навантажень доданий окремий тип дедуплікації, що вже включає в себе необхідні налаштування.

Підтримка Nano Server

Nano Server – це варіант розгортання операційної системи Windows Server 2025 з мінімальною кількістю встановлюваних компонентів. Nano Server повністю підтримує дедуплікацію.

Підтримка послідовного відновлення кластера

Послідовне відновлення кластера (Cluster OS Rolling Upgrade) – це нова функція Windows Server 2025, за допомогою якої можливо послідовно проапгрейдити операційну систему на кожному вузлі кластера з Server 2012 R2 до Server 2016, при цьому не припиняючи роботу кластера. Це можливо завдяки спеціальному змішаному режиму роботи кластера, коли в кластері одночасно можуть працювати вузли Windows Server 2025 R2 і Windows Server 2025.

Змішаний режим означає, що одні й ті-же дані можуть перебувати на вузлах з різними версіями дедуплікації. Дедуплікація Windows Server 2025

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

підтримує цей режим і забезпечує доступ до дедуплікованим даних протягом усього процесу відновлення кластера.

Установка й включення дедуплікації

Перше, що потрібно зробити для включення дедуплікації – це встановити відповідну роль сервера. Можливо з оснащення «Server Manager» запустити майстер і додати роль файлового сервера з компонентом «Data Deduplication».

Або з PowerShell виконати команду:

```
Install-WindowsFeature -Name FS-Data-Deduplication -  
IncludeAllSubfeature -IncludeManagementTools
```

Включення й налаштування дедуплікації

Після додавання компонента треба включити дедуплікацію для конкретного тому (або декількох томів). Зробити це можливо двома різними способами – із графічного оснащення або за допомогою PowerShell.

Для налаштування з GUI відкриваємо «Server Manager», переходимо в розділ «File and storage services» – «Volumes», встаємо на потрібний том, клацаємо по ньому правою клавішею миші й у меню, що відкрився, вибираємо пункт «Configure Data Deduplication».

Потім вибираємо необхідний тип дедуплікації й тиснемо «Apply». Додатково можливо вказати типи файлів, які не повинні піддаватися дедуплікації, а також виключити з дедуплікації певні директорії.

Наступне, що необхідно зробити після включення – це налаштувати розклад, по якому будуть відпрацьовувати завдання дедуплікації, тому далі тиснемо на кнопку «Set Deduplication Schedule».

За замовчуванням включена фонові оптимізація (background optimization), а також можливо налаштувати два додаткових завдання примусової оптимізації (throughput optimization). Налаштувань тут небагато – можливо тільки вибрати дні тижня, задати час запуску й тривалість роботи.

Набагато більше можливостей для налаштування надає PowerShell. Для включення дедуплікації використовуємо таку команду:

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

```
Enable-DedupVolume -Name D: -UsageType Hyper
```

Після включення переглянемо наявні завдання дедуплікації:

```
Get-DedupSchedule
```

Як бачите, тут картина трохи інша. Так крім фонові оптимізації є завдання пріоритетної оптимізації (PriorityOptimization), а також завдання складання сміття (GarbageCollection) і очищення (Scrubbing). Всі ці завдання не видні із графічного оснащення.

Крім цього, PowerShell дозволяє більш тонко налаштувати параметри виконання завдання. Для приклада створимо нове завдання оптимізації. Завдання повинне запускатися в 8 ранки з понеділка по п'ятницю й працювати 12 годин, з нормальним пріоритетом, використовувати не більше 50% оперативної пам'яті й 100% процесорної потужності:

```
New-DedupSchedule -Name ThroughputOptimization -Type Optimization -Days @(1,2,3,4,5) -DurationHours 12 -Start (Get-Date ?12/8/2016 8:00 PM?) -Memory 50 -Cores 100 -Priority Normal
```

Примітка. У принципі завданню можливо дати будь-яке ім'я. Але якщо створити завдання з ім'ям ThroughputOptimization, те воно буде відображатися й у графічному оснащенні.

Потім зробимо ще кілька змін. У перші перемінімо час і параметри запуску завдання збору сміття:

```
Set-DedupSchedule -Name WeeklyGarbageCollection -DurationHours 12 -Start (Get-Date ?12/8/2016 8:00 PM?) -Memory 50 -Cores 100 -Priority Normal -Full $true
```

Перенесемо на неділю завдання очищення:

```
Set-DedupSchedule -Name WeeklyScrubbing -Days Sunday -DurationHours 12 -Start (Get-Date ?12/8/2016 8:00 PM?) -Memory 50 -Cores 100 -Priority Normal
```

І відключимо пріоритетну оптимізацію:

```
Set-DedupSchedule -Name PriorityOptimization -Enabled $false
```

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Ручний запуск дедуплікації

При необхідності завдання дедуплікації можливо запустити вручну. Для приклада запустимо повну оптимізацію тому D з максимальним пріоритетом:

```
Start-DedupJob -Volume D: -Type Optimization -Memory  
75 -Cores 100 -Priority High -Full
```

Відстежити запущені завдання дедуплікації можливо за допомогою команди Get-DedupJob. Зверніть увагу, що одночасно відпрацьовує тільки одне завдання, а інші коштують у черзі й чекають її завершення. Якщо поточне завдання не завершиться найближчим часом, то завдання в черзі скасуються й черга буде очищена. Про це необхідно пам'ятати при складанні розкладу, тому що якщо різні завдання перетинаються між собою, те деякі з них не зможуть вчасно відробити.

Перегляд стану дедуплікації

Дані про стан дедуплікації для тому можливо подивитися командою:

```
Get-DedupVolume -Volume D: | fl
```

Так ми можемо подивитися основні параметри тому – повний обсяг, вільне місце, рівень стиску й т.д.

А перевірити стан завдань дедуплікації можливо такою командою:

```
Get-DedupStatus -Volume D: | fl
```

Ця команда крім усього іншого покаже, коли й з яким результатом були завершені завдання дедуплікації. У випадку проблем з дедуплікацією цю інформацію можливо використовувати для діагностики.

Вимикання дедуплікації

Відключити дедуплікацію для тому можливо або із графічного оснащення, забравши відповідну галочку, або за допомогою PowerShell. Наприклад:

```
Disable-DedupVolume -Name D:
```

Вимикання дедуплікації для тому скасовує всі заплановані завдання, а також забороняє запуск будь-яких завдань дедуплікації, крім read-only операцій (команди типу Get) і роздедуплікації (unoptimization). При цьому самі дані

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

залишаються в тому жє стані, у якому вони були до вимикання дедуплікації, просто нові дані перестають дедуплікуватися.

Після вимикання дедуплікації доступ до даних залишається й з ними можливо працювати так само, як і раніше. Це поведження можливо змінити, скориставшись ключем `DataAccess`, наприклад:

```
Disable-DedupVolume -Name D: -DataAccess
```

Ця команда видаляє драйвер-фільтр файлової системи для зазначеного тому, тим самим зупиняючи всі операції вводу-виводу для дедуплікованих файлів. Відповідно при спробі звертання до файлів ви одержите відмову в доступі. Повернути доступ можливо командою `Enable-DedupVolume` із ключем `DataAccess`.

Ну і якщо необхідно повністю забрати дедуплікацію й повернути дані до вихідного стану, то можливо скористатися процедурою роздедуплікації. Наприклад наступною командою запусимо роздедуплікацію для тому D з максимально-можливою швидкістю:

```
Start-DedupJob -Volume D: -Type Unoptimization -Memory 100 -Cores 100 -Priority High -Full
```

Зверніть увагу, що для роздедуплікації буде потрібно додатковий дисковий простір. Якщо вільного місця на томі недостатньо, то процедура завершиться помилкою. У цьому випадку можливо скористатися альтернативним способом – просто скопіювати дані на інший том. При копіюванні дані автоматично розтискаються, знову ж можливо роздедуплікувати не відразу весь том, а скопіювати дані вроздріб.

Завдання дедуплікації

Дедуплікація складається із трьох етапів, які виконуються у вигляді запланованих завдань:

– Оптимізація (Optimization) – у процесі оптимізації файли розбиваються на блоки (chunk), обчислюються співпадаючі блоки й зайві копії блоку видаляються, замінюючись посиланнями. Потім із блоків формуються

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

контейнери, які залежно від налаштувань, додатково стискаються й містяться в сховище блоків (chunk store).

– Складання сміття (Garbage Collection) – видалення блоків даних, на які немає активних посилань. При видаленні оптимізованого файлу стосовні до нього блоки не видаляються зі сховища негайно. Завдання складання сміття знаходить ці ?безхазяйні? блоки й видаляє їх, тим самим звільняючи дисковий простір.

– Очищення (Scrubbing) – перевірка цілісності дедуплікованих даних. Використовується для аналізу ушкоджень сховища й, по можливості, для відновлення ушкоджених даних.

Список всіх запланованих завдань можливо вивести командою:

```
Get-DedupSchedule
```

Завдання складання сміття й очищення додатково розділяються на звичайні й повні (full):

Повне складання сміття

При звичайному складанні сміття контейнер сховища блоків стискається тільки в тому випадку, якщо існує мінімальний відсоток блоків без посилань. При повнім складанні сміття контейнер стискається навіть у тому випадку, якщо на окремий блок у контейнері відсутня посилання. Також при повнім складанні звільняється місце, що могло бути використане при позаштатному завершенні роботи дедуплікації (напр. при збої живлення). Звичайний тип складання сміття працює швидше й споживає менше ресурсів, чим повний, але звільняє менше місця. Повне складання сміття звільняє до 5% більше місця, чим звичайна, але при цьому працює довше й споживає більша кількість системних ресурсів. За замовчуванням кожна 4-я процедура складання сміття є повною.

Повне очищення

Звичайне очищення перевіряє й виправляє тільки цілісність критичних метаданих і даних, для яких раніше були зафіксовані проблеми. При повнім очищенні перевірки піддаються все без винятку дані на томі. Цей тип очищення немає необхідності запускати часто, тому за замовчуванням він не

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

при його редагуванні командлетом Set-DedupSchedule. Давайте розглянемо деякі із цих параметрів:

- Type – тип завдання, може приймати одне із трьох значень (Optimization, GarbageCollection або Scrubbing).

- Days – дні тижня, на які запланований запуск завдання.

- Start – час запуску завдання.

- DurationHours – максимально припустимий час виконання завдання.

Після закінчення цього часу завдання буде зупинено в примусовому порядку, незалежно від того, чи відробило воно чи повністю ні.

- Memory – максимальний обсяг оперативної пам'яті, доступний для завдання. Задається у відсотках від загального обсягу пам'яті в системі.

- Cores – кількість ядер процесора (фізичн або логічних, якщо використовується HyperThreading), доступних для виконання завдання. Задається в процентному співвідношенні від загальної кількості ядер у системі.

- Priority – системний пріоритет, з яким буде запущене завдання. Можливо вказати низький (low), середній (normal) або високий (high) пріоритет. Залежно від пріоритету виробляється розподіл процесорної потужності в системі – чим вище пріоритет, тим більше процесорного часу одержить процес.

- StopWhenSystemBusy – указує, що дедуплікація повинна виконуватися тільки при наявності в системі вільних ресурсів. Якщо ж система зайнята й вільних ресурсів ні, дедуплікація повинна бути зупинена. За допомогою цього параметра можливо зменшити вплив дедуплікації на роботу сервера в тих випадках, коли вона виконується паралельно з іншими навантаженнями.

- InputOutputThrottle – рівень обмеження операцій вводу\виводу для завдання, число від 0 до 100. Дозволяє зменшити вплив дедуплікації на інші навантаження.

- InputOutputThrottleLevel – параметр, аналогічний попередній. Також служить для регулювання операцій вводу\виводу, може приймати значення Low,

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Normal або High. Якщо обидва параметри використовуються спільно, то InputOutputThrottle має більший пріоритет.

– Full – застосовується до завдань складання сміття й очищення, визначає, чи буде завдання повним або звичайним. Може приймати значення \$true або \$false.

– ReadOnly – застосовується до завдань очищення. При запуску очищення в цьому режимі виводиться інформація ушкоджених даних, але не вживають дії по їхньому виправленню.

Для наочності кілька прикладів. Спочатку створимо нове заплановане завдання:

```
New-DedupSchedule -Name ThroughputOptimization -Type Optimization
```

Як бачите, для нового завдання досить указати його ім'я й тип – єдині обов'язкові параметри. У цьому випадку буде створене щоденне завдання зі стандартними параметрами.

Обізнати завдання можливо як завгодно, але отут є цікавий момент. Для завдань продуктивної оптимізації існують два визначених імена: ThroughputOptimization і ThroughputOptimization-2. Ці імена даються завданням у тому випадку, якщо їх налаштувати їхнього оснащення Server Manager. І якщо при створенні нового завдання оптимізації з PowerShell указати одне із цих імен, то завдання буде відображатися й у графічному оснащенні.

Стандартний розклад може підходити не всім, тому змінимо його:

```
Set-DedupSchedule -Name ThroughputOptimization -Days @(1,2,3,4,5) -DurationHours 24 -Start ?8:00 PM?
```

Дні тижня можливо вказувати по іменах через кому (Sunday, Monday) або по номерах, у вигляді масиву чисел від 0 (неділя) до 6 (субота). Значення @(1,2,3,4,5) означає запуск завдання з понеділка по п'ятницю.

З параметром DurationHours є деякі незрозумілі моменти. При виставлянні його через графічне оснащення максимальне значення становить 24 години, при

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

налаштуванні через PowerShell обмеження становить 100 годин, а із планувальника – максимум 72 години (з дня). Ці ж 3 дні встановлюються при створенні завдання з PowerShell без вказівки даного параметра.

Параметр Start представляє із себе значення у форматі System.DateTime. Указувати його можливо також по різному, напр. **8:00 PM** означає старт в 8 годин вечора починаючи з поточного дня. Можливо вказати й дату повністю, напр. **Get-Date "12/8/2016 8:00 PM"**. При цьому сама дата не особливо важлива (якщо вона зазначена в минулому), а от частина із часом саме вказує на час запуску завдання. Також при вказівці дати треба враховувати увагу на формат часу, що використовується на сервері.

Ну й у завершення можливо підкоригувати параметри продуктивності для завдання, скажемо для зменшення впливу дедуплікації на роботу сервера:

```
Set-DedupSchedule -Name ThroughputOptimization -  
Memory 25 -Cores 75 -Priority Low -InputOutputThrottle 20  
-StopWhenSystemBusy $true
```

Всі завдання дедуплікації можливо побачити в планувальнику (Task scheduler), у розділі Microsoft\Windows\Deduplication.

Там також можливо налаштувати розклад завдань. Більше того, деякі параметри можливо змінити тільки за допомогою планувальника. Наприклад завдання пріоритетної й фонові оптимізації запускається в момент включення дедуплікації для тому, після чого відпрацьовують щогодини в плинні необмеженого часу. Ці налаштування не можливо побачити\змінити за допомогою засобів керування дедуплікацією, вони доступні тільки в планувальнику.

Також планувальник дозволяє більш гнучко управляти розкладом дедуплікації. З його допомогою можливо скласти не щотижневе, а щомісячний розклад, наприклад кожну другу неділю місяця.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Однак при налаштуванні щомісячного розкладу будинок пропаде зі списку і його буде неможливо відредагувати. Для редагування прийде спочатку повернутися до стандартного розкладу.

Налаштування дедуплікації для тому

Деякі параметри настраюються на рівні тому. Переглянути їх можливо за допомогою командлета Get-DedupVolume. Наприклад:

```
Get-DedupVolume D: | fl
```

Крім іншого ця команда виводить деякі важливі налаштування, які можуть впливати на продуктивність дедуплікації:

– `MinimumFileAgeDays` – вік файлу, по досягненні якого файл вважається придатним для оптимізації. За замовчуванням для файлових серверів і віртуальних машин Hyper-V цей параметр дорівнює 3, тобто файли молодше 3 днів не піддаються оптимізації. Зменшення цього значення здатно підвищити ефективність дедуплікації, однак це може негативно позначитися на продуктивності системи.

– `MinimumFileSize` – мінімальний розмір файлу, що вважається придатним для оптимізації. За замовчуванням файли розміром менше 32кб не оптимізуються, оскільки їхня оптимізація не приносить відчутної економії дискового простору.

– `NoCompress` – параметр вказує, чи варто стискати блоки перед їхнім приміщенням у сховище. За замовчуванням стиск включений, але в деяких випадках, наприклад при оптимізації вже стислих даних (напр. архіви) стиск можливо відключити. Це дозволить знизити навантаження, створювану дедуплікацією.

– `NoCompressionFileType` – типи файлів, блоки яких не треба стискати перед приміщенням у сховище. Цей параметр являє собою масив розширень файлів. Якщо попередній параметр виключає стиск для всього тому, то тут ми прицільно можемо задати типи файлів, які немає необхідності стискати,

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

наприклад аудіо\ відео-файли або стислі архіви. Деякі з них уже включені в масив за замовчуванням.

– `OptimizeInUseFiles` – визначає, чи можливо вважати доступними для оптимізації відкриті файли. Цей параметр повинен бути включений при дедуплікації файлів, які залишаються відкритими тривалий час, таких як VHD-диски працюючих віртуальних машин або файли баз даних. Без нього такі файли не будуть оптимізовані.

– `OptimizePartialFiles` – оптимізація файлів вроздріб. При включенні цього параметра при оптимізації параметр `MinimumFileAgeDays` застосовується не до всього файлу, а до його окремих сегментів. Цей параметр повинен бути включений у тому випадку, якщо оптимізація застосовується до більших, що постійно змінюються файлам, у яких змінюється тільки невелика частина, а більша частина файлу залишається без змін.

– `InputOutputScale` – задає рівень паралелізації операцій вводу\виводу в процесі оптимізації тому. Припустимі значення від 0 до 36. При значенні 0 (значення за замовчуванням) система автоматично визначає рівень паралелізації залежно від розміру тому. Цей параметр новий, він з'явився в Windows Server 2025 і відповідає за обмеження кількості черг вводу\виводу при обробці даних. Перед його зміною варто добре подумати, тому що неоптимальне значення може дуже сильно зменшити швидкість роботи дедуплікації.

– `Verify` – відповідає за додаткову перевірку цілісності. При включенні цього параметра якщо геш блоку відповідає блоку, уже наявному в сховище, блоки рівняються побайтно. Теоретично це повинне виключити помилку при збігу гешу у двох різних блоків, на практиці ж така ситуація вкрай мало ймовірна. Крім того, включення функції перевірки істотно підвищує навантаження на систему.

– `ChunkRedundancyTreshold` – кількість посилок на блок даних, при досягненні якого блок необхідно продублювати у розділі активної зони сховища. У такий спосіб прискорюється доступ до блоків, на які часто посилаються. За

замовчуванням цей параметр дорівнює 100 і без гострої потреби його краще не міняти, тому що це може понизити ефективність дедуплікації.

Змінити параметри дедуплікації для тому можливо за допомогою командлета Set-DedupVolume. Для приклада зменшимо мінімальний вік файлу до нуля й відключимо часткову дедуплікацію файлів для тому:

```
Set-DedupVolume -Volume D: -MinimumFileAgeDays 0 -  
OptimizePartialFiles:$false
```

Типи дедуплікації

Тепер плавно переходимо до вибору типу дедуплікації. Тип дедуплікації являє собою готовий набір налаштувань, оптимізований під певне робоче навантаження. Це дозволяє не заморачуватися з підбором параметрів, а просто вибрати підходящий тип дедуплікації. В Windows Server 2025 на вибір пропонується 3 типи:

Default

Цей тип призначений для файлових серверів загального призначення. Для нього включена фонові оптимізація й використовуються наступні налаштування:

- MinimumFileAgeDays – 3 дні;
- OptimizeInUseFiles – \$false;
- OptimizePartialFiles – \$false.

Hyper

Цей тип застосовується для дедуплікації віртуальних машин Hyper-V. Для нього крім фонові включена пріоритетна оптимізація й використовуються наступні налаштування:

- MinimumFileAgeDays – 3 дні;
- OptimizeInUseFiles – \$true;
- OptimizePartialFiles – \$true.

Backup

Тип дедуплікації, спеціально призначений для віртуалізованих серверів резервного копіювання, таких як Microsoft Data Protection Manager. Для цього

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

типу також за замовчуванням включені фонові й пріоритетна оптимізація, налаштування наступні:

- `MinimumFileAgeDays` – 0 днів;
- `OptimizeInUseFiles` – `$true`;
- `OptimizePartialFiles` – `$false`.

У документації Microsoft згадується про додаткові тонкі налаштування, які застосовуються для кожного конкретного типу використання. Можливо ці налаштування сховані від користувача, тому що при порівнянні різних типів дедуплікації я не знайшов істотних розходжень, крім описаних вище.

Налаштування операційної системи

Деякі налаштування можливо змінити на рівні операційної системи, за допомогою реєстру. Налаштування дедуплікації можливо знайти у розділі **HKLM\System\CurrentControlSet\Services\ddpsvc\Settings** для одиночного сервера або в **HKLM\Cluster\Dedup** для кластера. Ці параметри застосовуються до усім без винятку томам і завданням дедуплікації, що виконуються в системі.

Приміром, для налаштування інтервалу запуску завдання повного складання сміття можливо додати параметр `DWORD` з ім'ям **DeepGCInterval** такою командою:

```
Set-ItemProperty -Path  
HKLM:\System\CurrentControlSet\Services\ddpsvc\Settings -  
Name DeepGCInterval -Type Dword -Value 0
```

Значення цього параметра задає інтервал запуску повного складання сміття, тобто кожне N-Е завдання є завданням повного складання сміття. За замовчуванням повне складання сміття запускається кожне четверте завдання, але іноді це потрібно змінити, наприклад збільшити цей інтервал. А якщо задати параметру значення 0, те повне складання сміття буде зовсім відключена.

Для виправлення ситуації, коли завдання дедуплікації відміняється з помилкою «uses too much memory» можливо додати параметр `DWORD` з ім'ям **WlmMemoryOverPercentThreshold**, наприклад:

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

```
Set-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\ddpsvc\Settings -Name WlmMemoryOverPercentThreshold -Type Dword -Value 300
```

Значення параметра означає, наскільки завдання дедуплікації може збільшити встановлений по пам'яті ліміт. Наприклад, значення 300 означає, що що можливо перевищити ліміт в 3 рази (на 300%).

При дуже високому рівні стиску (saving rate>80%) завдання оптимізації можуть відпрацьовувати повільніше із за той, що потрібно виділяти більше пам'яті для обробки стиску. Для коректування обсягу пам'яті можливо додати параметр **EstimatedCompressionRate** такою командою:

```
Set-ItemProperty -Path HKLM:\System\CurrentControlSet\Services\ddpsvc\Settings -Name EstimatedCompressionRate -Value 5
```

Значення параметра встановлюється в діапазоні від 5 до 10, залежно від рівня стиску. Для понад-високий рівень стиску (90%-95%) рекомендується значення 10.

Налаштування файлової системи

У деяких випадках для коректної роботи дедуплікації може знадобитися спеціальним образом підготувати файлову систему. Так при використанні більших (1ТБ і більше) файлів рекомендується задати розмір кластера 64кб, а також збільшити розмір сегмента запису файлу (File Record Segment, FRS) до 4кб.

За замовчуванням розмір кластера NTFS становить 4кб, а розмір FRS – 1кб. Перевірити ці значення можливо за допомогою утиліти fsutil, командою:

```
fsutil fsinfo ntfsinfo D
```

Для зміни цих параметрів буде потрібно переформатувати том. Зробити це можливо з командного рядка:

```
format D: /A:64K /L /Q
```

Або за допомогою PowerShell:

```
Format-Volume -Partition D: -FileSystem NTFS -AllocationUnitSize 64KB -UseLargeFRS -Force
```

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Ці налаштування особливо необхідні для таких навантажень, як віртуальні сервера резервного копіювання. Без них можлива ситуація, коли при інтенсивних навантаженнях сервер втрачає диски. Нагадаю, що ці маніпуляції необхідно проводити на порожньому томі, оскільки при форматуванні всі дані будуть знищені.

Моніторинг і перевірка стану

Для перегляду й аналізу я волю використовувати командлети `Get-DedupStatus` або `Update-DedupStatus`, які показують всі основні дані про стан томи – зекономлене місце, рівень стиску, кількість оптимізованих файлів і т.п.

Обидва командлети виводять однаковий набір даних, але між ними є одна відмінність. `Get-DedupStatus` показує закешировані дані, тоді як `Update-DedupStatus` перед виводом робить сканування тому на предмет змін.

Крім того, вони виводять дані про час запуску й результат виконання завдань дедуплікації. У випадку проблем з дедуплікацією ця інформація допоможе знайти причину. Так у нашій прикладі можливо побачити, що останнє завдання складання сміття не змогло відробити до кінця й було скасовано.

Більше докладну інформацію можливо знайти в системному журналі, у розділі «Application and Services Log\Microsoft\Windows\Deduplication». Як бачите, у нашій випадку завдання було скасовано вручну.

Ну й для розширеного аналізу можливо використовувати дані, видавані командлетом `Get-DedupMetadata`. Він повертає докладні дані про стан сховища дедуплікованих даних:

- `TotalChunkStoreSize` – розмір сховища блоків;
- `DataChunkCount` – число блоків даних на томі;
- `DataChunkAverageSize` – середній розмір блоку. Обчислюється як розмір сховища, ділений на загальне число блоків;
- `DataContainerCount` – число контейнерів у сховище;
- `HotspotCount` – число блоків в активній зоні. В активну зону попадають блоки, на які часто посилаються (за замовчуванням більше 100 разів). Всі блоки в

активній зоні дублюються, щоб забезпечити більше швидкий доступ і автоматичне відновлення у випадку їхнього ушкодження;

- HotspotContainerCount – число контейнерів в активній зоні;
- CorruptionsCount – число ушкоджених елементів, знайдених на томі;
- CorruptionsLogEntryCount – число записів про ушкоджені елементи на томі.

– StreamMapCount – число схем потоків даних на томі. Наскільки я розумію, під схемою потоку розуміється набір метаданих, що визначають розташування блоків з даним;

- StreamMapChunkCount – число блоків у сховище схем потоків;
- StreamMapContainerCount – число контейнерів у сховище схем потоків.

Так у нашій прикладі загальний розмір сховища становить близько 33.5 Гб. Дедуплікації піддалися 501758 блоків, середній розмір блоку становить приблизно 70 кб.

Висновок

На закінчення опишу деякі проблеми, з якими можливо зіткнутися при використанні дедуплікації.

Навантаження

При використанні дедуплікації треба розуміти, що процедура ця вимагає досить багато обчислювальних ресурсів процесора, пам'яті й дискових операцій. Особливо вимоглива дедуплікація до оперативної пам'яті, для обробки 1 Тб даних потрібно приблизно 2 Гб пам'яті. Звичайно ця цифра зразкова й залежить від безлічі факторів – типу навантаження, частоти зміни даних і т.п.

При недостатчі ресурсів дедуплікація починає працювати повільно й неефективно (збільшується час виконання, зменшується рівень стиску) і навіть може повністю зупинитися. З іншого боку, дедуплікацію все-таки необхідно обмежувати, тому що в протилежному випадку вона може вибрати всі вільні ресурси й покласти сервер.

Розклад

В ідеалі розклад повинне бути складене таким чином, щоб дедуплікація вироблялася у позаробочій час, щоб рознести дедуплікацію за часом з основним навантаженням. Однак це можливо далеко не завжди. На більших обсягах даних, особливо якщо дані часто змінюються, завдання дедуплікації може працювати кілька днів. Якщо ж штучно обмежувати тривалість роботи завдання, то ефективність дедуплікації може помітно впасти.

Другий момент, пов'язаний з розкладом – це необхідність регулярного запуску завдань складання сміття й очищення. Проблема в тому, що всі завдання виконуються по черзі, і якщо на момент запуску одного завдання виконується інше, те нове завдання ставиться в чергу й через якийсь час відміняється. У результаті завдання складання й очищення можуть не виконуватися протягом тривалого часу, що приводить до сумних наслідків.

При відсутності регулярного складання сміття збільшується займане місце, а без очищення можливі ушкодження й втрата даних. Тому бажано скласти розклад таким чином, щоб на момент їхнього запуску в системі не було запущених завдань дедуплікації.

Підсумовуючи все вищесказане – при налаштуванні дедуплікації необхідно знайти баланс, при якому всі завдання дедуплікації будуть ефективно відпрацьовувати, при цьому впливаючи на роботу інших сервісів, що перебувають на сервері. Як з'ясовується на практиці, підібрати навантаження й виставити оптимальні параметри можливо тільки досвідченим шляхом.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення написане мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-орієнтована мова, призначена для розробки різноманітних безпечних і потужних

					VKPM-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

застосунків, виконуваних у середовищі .NET Framework. Мовою Visual C# можливо розробляти звичайні клієнтські додатки Windows, веб-служби XML, розподілені компоненти, додатки типу “сервер-клієнт”, додатки баз даних і багато яких інших. В Visual C# є розширений редактор коду, конструктори зі зручним користувальницьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку застосунків 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 застосунку – 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# можливо 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 є міжнародним стандартом, який лежить в основі створення середовищ

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

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, створений компілятором Visual C# відповідає специфікації CTS, код IL на основі Visual C# може взаємодіяти з кодом, створеним 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. Крім служб часу виконання, в .NET Framework також є велика бібліотека, що складається з більш ніж 4000 класів, організованих по просторах імен, які забезпечують різноманітні корисні функції для будь-яких дій, починаючи від введення й виведення файлів для керування рядками для розбивки XML, і закінчуючи елементами керування Windows Forms.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи дедуплікації в інфраструктурі резервного копіювання.

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-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Дедуплікація буває різна – на рівні файлів, блоків даних і навіть на бітовому рівні. Використовується блокова дедуплікація. Файли розбиваються на невеликі блоки різного розміру (32-128 КБ), визначаються дублюючі блоки й зберігається одна копія кожного блоку. Надлишкові копії блоку замінюються посиланнями на цю єдину копію. Блоки організуються у файли-контейнери, які можуть стискуватися для подальшої оптимізації використання простору, і містяться в сховище блоків.

Для приклада припустимо, у нас є два файли – File1 і File2. У вихідному стані вони містять метадані (ім'я файлу, атрибути й т.п.) і самі дані.

Після дедуплікації дані з File1 і File2 видаляються й замінюються заглушками, що вказують на відповідні блоки даних, що зберігаються в загальному сховищі блоків. Тому що блоки А, В і С однакові для обох файлів, вони зберігаються в єдиній копії, що знижує обсяг дискового простору, необхідний для зберігання обох файлів.

Під час доступу до одному з файлів відповідні блоки збираються разом. При цьому користувач або додаток працюють із файлом як і раніше, не підозрюючи про те, що файл був підданий перетворенням. Це дозволяє застосовувати дедуплікацію, не турбуючись про її вплив на поведження застосунків або доступ користувачів до файлу.

Таким чином, після включення дедуплікації тому й оптимізації даних том містить:

– Оптимізовані файли (файли точок повторного аналізу) які містять покажчики на відповідні блоки даних у сховище блоків, необхідні для побудови вихідного файлу.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

- Сховище блоків (дані оптимізованих файлів).
- Неоптимізовані файли (тобто пропущені файли, наприклад файли стану системи, зашифровані файли, файли з додатковими атрибутами або файли розміром менш 32 КБ).

Планування

Дедуплікація може значно знизити споживаний дисковий простір (на 50-90% і більше), але тільки при правильному плануванні. Тому при виборі об'єкта для дедуплікації варто враховувати деякі моменти.

Тип даних

Ефективність дедуплікації дуже сильно залежить від типу даних. Так мультимедійні файли (фотографії, музика, відео) практично не містять повторюваних даних, тому їх дедуплікація не дасть великої економії. У той же час файли віртуальних машин (VHD) чудово дедуплікуються й на них економія може становити до 95 %. Із цієї причини перед включенням дедуплікації рекомендується виконати попередню оцінку даних на предмет дедуплікуємості.

Частота зміни файлів

Файли, які часто змінюються й до яких часто звертаються користувачі або додатки, не дуже підходять для дедуплікації. Постійний доступ до даних і їхній зміна швидше за все зведуть нанівець всі результати дедуплікації й можуть просто не дати дедуплікації можливості обробити файли. Простіше говорячи, для дедуплікації добре підійдуть дані, які часто читають, але рідко змінюють.

Завантаженість сервера

Під час дедуплікації виконується читання, обробка й запис великого обсягу даних. Цей процес споживає ресурси сервера, що необхідно враховувати при плануванні розгортання. Як правило, сервера мають періоди високої й низької активності. Більшу частину дедуплікації можливо виконати, коли ресурси доступні. Постійно високонагруженні сервера не рекомендується використовувати для дедуплікації.

Не рекомендується виконувати дедуплікацію файлів, які відкриті, постійно змінюються протягом тривалого періоду часу або мають високі вимоги вводу/виводу, наприклад файли працюючих віртуальних машин, динамічних баз даних SQL або активних сеансів VDI. Справа в тому, що при дедуплікації не виконується обробка файлів, відкритих постійно в монопольному режимі для запису. Це значить, що дедуплікація не буде проведена доти, поки файл не буде закритий. Тільки тоді завдання оптимізації виконає спробу обробити файл, що відповідає обраним параметрам політики дедуплікації.

У принципі дедуплікацію можливо налаштувати на обробку постійно, що змінюються файлів. Але в цьому випадку можлива ситуація, коли процес оптимізації не зможе одержати доступ до цих файлів і пропустить їхню обробку. Не варто витрачати ресурси сервера на дедуплікацію файлів, у які постійно записуються нові дані.

Приведу рекомендації Microsoft. Для дедуплікації:

Не рекомендується

- Сервера Hyper-V.
- VHD-Файли запущених віртуальних машин.
- Служби WSUS.
- Сервера SQL і Exchange.
- Будь-які файли, розмір яких дорівнює або більше 1 Тб.

Рекомендується:

- Файлові ресурси загального доступу (загальні папки, профілі й домашні папки користувачів, інші файлопомойки).
- Розгортання програмних продуктів (бінарники, образи дисків і відновлення ПЗ).
- Бібліотеки віртуалізації (VHD-диски).
- Томи архівів SQL і Exchange.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Треба сказати, що рекомендації Microsoft часто суперечать один одному, тому не треба їх беззастережно приймати на віру. У кожному разі перед включенням дедуплікації необхідний ретельний аналіз.

Для визначення очікуваної економії в результаті включення дедуплікації можливо використовувати засіб оцінки дедуплікації Ddpeval.exe. Після установки компонента дедуплікації утиліта Ddpeval.exe автоматично встановлюється в папку \Windows\System32\. До речі, її можливо просто скопіювати з будь-якої установки Windows Server 2025 і запустити в системах Windows 10/11.

Синтаксис у програми простіше нікуди, пишемо Ddpeval.exe і вказуємо шлях. Як шлях можливо вказати локальний диск, папку або мережну кулі:

```
Ddpeval.exe E:
```

```
Ddpeval E:\Test\
```

```
Ddpeval.exe \\Server\Share
```

Програма видасть очікуваний розмір економії дискового простору, після чого вже можливо приймати рішення – включати дедуплікацію чи ні.

Системні вимоги

Дедуплікація пред'являє до системи деякі вимоги.

Тому

– Тома, призначені для дедуплікації не повинні бути системними або завантажувальними. Дедуплікація не підтримується для томів операційної системи.

– Тома можуть бути розбиті під MBR або GPT і відформатовані в NTFS. Нова відказостійка файлова система ReFS не підтримується.

– Тома можуть перебувати на локальних дисках або в загальнодоступному сховищі (SAS, iSCSI або Fibre Channel).

– Windows повинна бачити томи як незнімні диски. Мережні диски й знімні носії не підтримуються.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

– Не можливо включати дедуплікацію для загальних томів кластера (Cluster Shared Volume, CSV). Якщо дедуплікований том перетворити в CSV, то доступ до даних залишиться, але завдання дедуплікації не зможуть відробити.

Апаратні ресурси

– Устаткування серверів повинне відповідати мінімальним вимогам Windows Server 2025. Функція дедуплікації розроблена для підтримки мінімальних конфігурацій, таких як система з одним процесором, 4 ГБ ОЗУ й одним жорстким диском SATA.

– Сервер повинен мати одне процесорне ядро й 350 МБ вільної пам'яті для виконання завдання дедуплікації на одному томі, при цьому буде оброблятися близько 1,5 ТБ даних у день. Якщо планується підтримувати дедуплікацію в декількох томах на одному сервері, необхідно відповідним чином збільшити продуктивність системи, щоб гарантувати, що вона зможе обробляти дані.

– Функція дедуплікації підтримує одночасну обробку до 90 томів, однак при дедуплікації одночасно може оброблятися один том на фізичне процесорне ядро плюс один. Застосування технології Hyper-Threading не впливає на цей процес, оскільки для обробки тому можливо використовувати тільки фізичні ядра. Наприклад сервер з 16 процесорними ядрами й 90 томами буде обробляти по 17 томів одночасно, поки не обробить всі 90 томів.

– До віртуальних серверів застосовуються ті ж правила, що й до фізичного встаткування відносно ресурсів сервера.

Загальні вимоги

– Наявність вільного місця на диску. При відсутності дискового простору на дедуплікованому томі деякі додатки не зможуть одержати доступ до даних і будуть завершені з помилкою. Необхідно зберігати, принаймні, один гігабайт вільного місця на дедуплікованому томі.

– Тверді квоти. При використанні FSRM (File System Resource Manager) не підтримується установка твердих квот на обсяг тому. Коли для тому встановлені

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

тверді квоти, фактичний обсяг вільного місця на томі й обмежене квотами простір відрізняється, що може привести до невдачі процесу дедуплікації. Всі інші FSRM-Квоти, у тому числі м'які квоти на обсяг тому й квоти на підпапки, будуть нормально працювати при дедуплікації.

– Файли з додатковими атрибутами, зашифровані файли, файли розміром менше 32 КБ і файли точок повторного аналізу при дедуплікації не обробляються.

3.2 Розробка структурної схеми

Дедуплікація, по визначенню IDC, – це технологія створення з дублікатів єдиної копії даних з можливістю спільного доступу, що підвищує ефективність використання ємності систем зберігання. Згідно Microsoft, це процедура пошуку й видалення даних, що дублюються, без шкоди для їхньої якості й цілісності з метою зменшення обсягу простору, займаного даними.

Блоки, загальні для декількох файлів, зберігаються у вигляді однієї копії, тому вимоги до ємності зберігання всіх файлів знижуються. Видаляючи повторювані послідовності даних, дедуплікація дозволяє значно скоротити обсяг переданих і/або збережених даних. Щоб ще сильніше його зменшити, дедуплікацію нерідко сполучать із компресією (стиском), називаючи все це ущільненням даних.

При запиті файл збирається з відповідних блоків (для користувача або застосунку цей процес прозорий), тому, застосовуючи дедуплікацію до файлів, можливо не турбуватися про те, що робота застосунків ускладниться або доступ користувачів до файлів виявиться неможливий. Дедуплікація знижує витрати, зводячи до мінімуму накладні витрати на зберігання й передачу даних.

При використанні в системах резервного копіювання дедуплікація допомагає вирішити цілий ряд проблем і дає вагомі переваги.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Основні причини використання дедуплікації даних у системі резервного копіювання – невеликий розмір резервних копій, зниження потреби в ємності зберігання, скорочення мережного трафіку, зменшення вікна резервного копіювання. Щодня виконуване повне резервне копіювання дозволяє гарантувати швидке відновлення за один крок.

У системах резервного копіювання з функцією дедуплікації можливі різні варіанти зберігання резервних копій: стрічкові накопичувачі й бібліотеки, віртуальні стрічкові бібліотеки, дискові масиви й системи зберігання даних, у тому числі з убудованої дедуплікацією. Дедуплікація виконується «на лету» (у процесі резервного копіювання або архівування), тому на диску, а також у хмарних сховищах (cloud backup) зберігаються вже дедупліковані дані.

Вимоги до ємності скорочуються в середньому 10-30 разів, значно підвищуються швидкість і надійність відновлення й добування даних. Але ніщо не дається даром: дедуплікація вимагає витрат. Потрібні обчислювальні ресурси, до того ж відбувається зниження продуктивності систем або збільшується їхня вартість. Тому потрібно з'ясувати, які витрати неминучі при використанні кожного методу.

Існують різні види, або методи, дедуплікації, у кожного свої переваги й недоліки.

Дедуплікація може виконуватися на джерелі даних або безпосередньо в сховище. Залежно від цього розподіляється обчислювальне навантаження. При першому методі дані обробляються на стороні клієнта й по закінченні дедуплікації пересилаються на пристрої зберігання. У результаті навантаження на мережу знижується, але на клієнтах доводиться встановлювати спеціалізоване ПЗ й оснащувати їх значними обчислювальними ресурсами.

Другий метод передбачає застосування могутнішої й дорогих СЗД, але витрати на первісну передачу даних фактично не знижуються. У кожному разі потрібно враховувати, що вимоги дедуплікації до ресурсів системи досить високі.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

По даним Dell EMC, дедуплікація на джерелі дозволяє скоротити ємність зберігання до 50 разів, трафік – до 500 разів, час резервного копіювання – до 10 разів. Однак її ефективність дуже сильно залежить від типу даних. Очевидно, що найбільший ефект досягається, коли дані мають велику надмірність, а також коли копіюються й/або зберігаються після внесення незначних змін.

У загальному випадку для неструктурованих даних (файли документів, журналів, образів і віртуальних машин, електронної пошти й архівів) характерний високий коефіцієнт дедуплікації – їхній обсяг нерідко зменшується в 20–30 разів. Наприклад, добре дедуплікуються файли віртуальних машин (VHD): економія може становити до 90%. Дедуплікація структурованої інформації, наприклад баз даних, не настільки ефективна (звичайно до 5-8 разів).

Дедуплікація мультимедійних файлів теж не забезпечує великої економії. Зображення, відео (JPEG, GIF, TIF, MPEG, і ін.), результати стиску й шифрування, картографічні й сейсмічні дані, файли САПР – всі ці формати проблемні для дедуплікації.

Добре дедуплікуються рідко змінювані дані, оскільки постійний доступ до даних і їхній зміну можуть звести до мінімуму ефект дедуплікації. Microsoft не рекомендує застосовувати дедуплікацію до серверів Hyper-V, SQL і Exchange, файлам запущених віртуальних машин, більшим файлам (більше терабайта). Вона може функціонувати на файловому, блоковому або бітовому рівні.

У деяких схемах, заснованих на ґешуванні, для підвищення коефіцієнта дедуплікації застосовується попередня обробка даних. Якщо дані обробляються на стороні клієнта, дедуплікація блоків змінної довжини дозволяє значно зменшити час резервного копіювання, тому що зберігаються тільки унікальні сегменти. Цей метод більше ефективний, чим традиційна дедуплікація сегментів фіксованої довжини, коли навіть невеликі зміни в наборі даних приводять до резервного копіювання всього файлу.

У системах резервного копіювання можуть використовуватися й інші методи дедуплікації. До того ж розроблювачі систем зберігання застосовують

різні алгоритми дедуплікації, у тому числі досить складні, для яких потрібно більше процесорних ресурсів. Тому величина коефіцієнта дедуплікації залежить від реалізації цієї технології.

Сучасні рішення дають можливість створювати резервні копії даних у хмарі, що дозволяє надійно зберегти й відновити їх, визволивши час і ресурси.

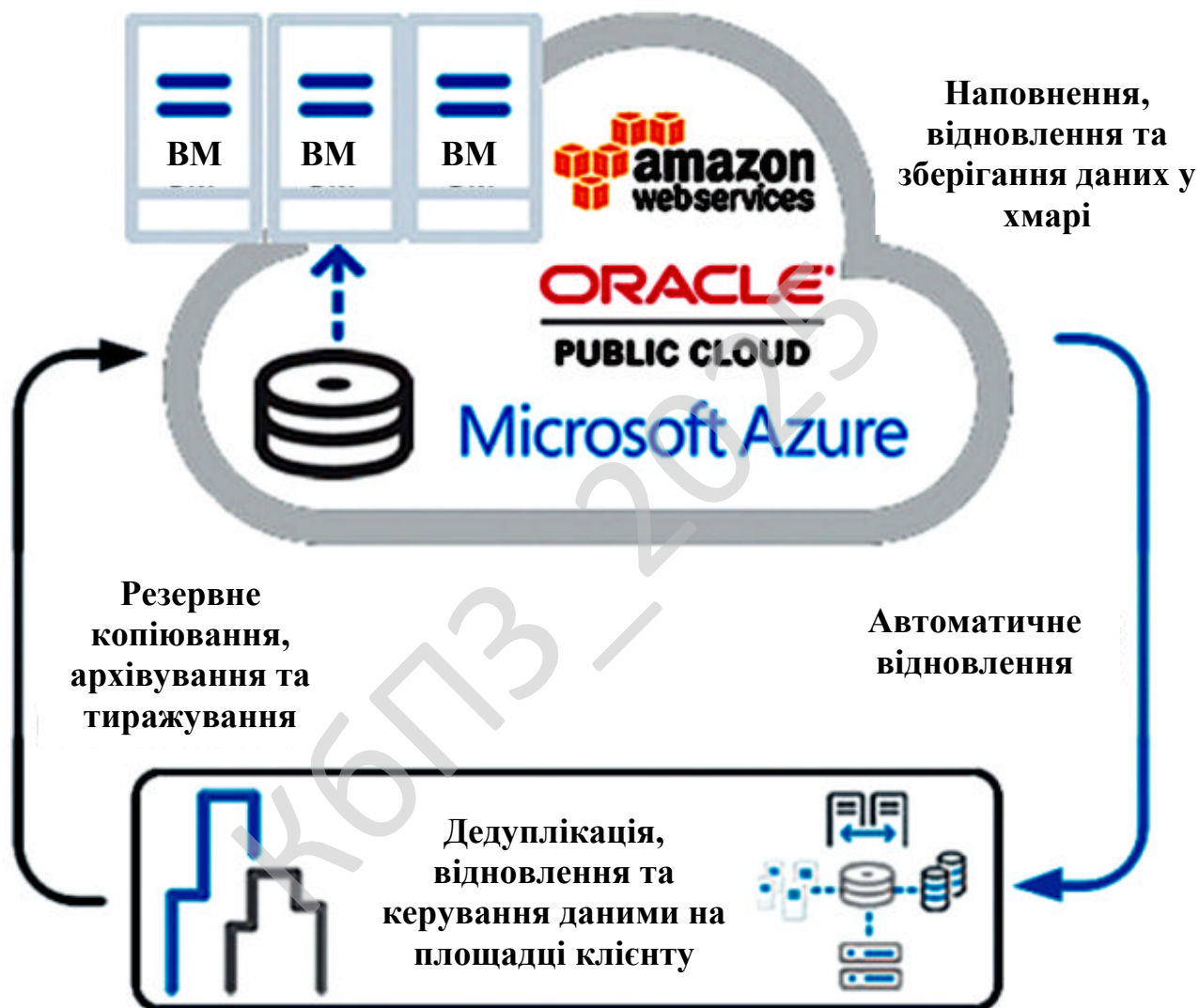


Рисунок 3.1 – Структурна схема системи

Вихідні дані можуть перебувати на будь-якій площадці: на стороні клієнта, в іншому комерційному ЦОДі або в хмарі. Таким чином, можливо швидко реалізувати переваги резервного копіювання даних у хмару,

використовуючи його як розширення корпоративної IT-інфраструктури. У числі переваг такого рішення:

- швидке й надійне резервне копіювання, оперативне відновлення даних;
- відсутність необхідності в дорогих шлюзах і складних проміжних рішеннях;
- можливість замінити стрічкове сховище хмарним;
- використання хмари для аварійного відновлення.

Компанії все частіше застосовують комбінований підхід: дискові масиви в сполученні з копіюванням у хмару. Така стратегія зберігання резервних копій дозволяє прискорити резервне копіювання найбільш актуальних даних і знизити вартість володіння відповідною інфраструктурою резервного копіювання. Використання дедуплікації дає можливість не тільки оптимізувати займану резервними копіями ємність сховищ, але й зменшити мережний трафік при резервуванні в хмару, що робить цю технологію ще більш актуальною.

Постійний розвиток і інтеграція рішень дедуплікації даних розроблювачами систем резервного копіювання й виробниками систем зберігання буде сприяти більшій економічності пропонованих рішень, зниженню вартості впровадження, більше швидкому переміщенню резервних копій між територіально віддаленими площадками.

3.3 Розробка функціональної схеми

У даній роботі, для системи дедуплікації в інфраструктурі резервного копіювання застосовується, крім усього вище перерахованого, також технологія RAID-масивів. Відразу варто помітити, що RAID розшифровується як Redundant Array of Independent Disks – надлишковий масив незалежних дисків. Споконвічно RAID розшифровувався як Redundant Array of Inexpensive Disks – надлишковий масив недорогих дисків. Під недорогими малися на увазі диски, призначені для використання в ПК, – на противагу дорогим дискам для мейнфреймів. Але так як

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

в RAID-масивах стали використовувати SCSI-вінчестери, які були істотно дорожче застосовуваних у більшості комп'ютерів дисків IDE, слово "недорогий" було замінено на "незалежний".

Принцип функціонування RAID-системи полягає в наступному: з набору дискових накопичувачів створюється масив, що управляється спеціальним контролером і визначається комп'ютером як єдиний логічний диск великої ємності. За рахунок паралельного виконання операцій вводу-виводу забезпечується висока швидкодія системи, а підвищена надійність зберігання інформації досягається дублюванням даних або обчисленням контрольних сум. Слід зазначити, що застосування RAID-масивів захищає від втрат даних тільки у випадку фізичної відмови жорстких дисків.

Розрізняють кілька основних рівнів RAID-масивів: RAID 0, 1, 2, 3, 4, 5, 6, 7. Також існують комбіновані рівні, такі як RAID 10, 0+1, 30, 50, 53 і т.п. Розглянемо принципи функціонування, достоїнства й недоліки основних рівнів.

RAID 0 – Дисковий масив без відказостійкості (Striped Disk Array without Fault Tolerance)

Дисковий масив без надлишкового зберігання даних. Інформація розбивається на блоки, які одночасно записуються на окремі диски, що забезпечує підвищення продуктивності. Такий спосіб зберігання інформації ненадійний, оскільки поломка одного диска приводить до втрати всієї інформації, тому рівнем RAID як таким не є.

RAID 0 – дешевий і продуктивний, але ненадійний. За рахунок можливості одночасного вводу/виводу з декількох дисків масиву RAID 0 забезпечує максимальну швидкість передачі даних і максимальну ефективність використання дискового простору, тому що не потрібно місця для зберігання контрольних сум. Реалізація цього рівня дуже проста. RAID 0, як правило, застосовується в тих областях, де потрібна швидка передача великого обсягу даних. Для реалізації масиву потрібно не менше двох вінчестерів.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Переваги:

- найвища продуктивність у додатках, що вимагають інтенсивної обробки запитів вводу/виводу й даних великого обсягу;
- простота реалізації;
- низька вартість;
- максимальна ефективність використання дискового простору – 100%.

Недоліки:

- не є "сьогоденням" RAID'ом, оскільки не підтримує відказостійкість;
- відмова одного диска спричиняє втрату всіх даних масиву.

RAID 1 – Дисковий масив із відзеркалюванням (Mirroring & Duplexing)

Дисковий масив з дублюванням інформації (відзеркалюванням даних). У найпростішому випадку два накопичувачі містять однакову інформацію і є одним логічним диском. При виході з ладу одного диска його функції виконує інший. Для реалізації масиву потрібно не менше двох вінчестерів.

RAID 1 – найпростіший відказостійкий масив.

Переваги:

- простота реалізації;
- простота відновлення масиву у випадку відмови (копіювання).

Недоліки:

- висока вартість – 100-процентна надмірність;
- невисока швидкість передачі даних.

RAID 2 – Відказостійкий дисковий масив з використанням коду Хеммінга (Hamming Code ECC)

Схема резервування даних з використанням коду Хеммінга (Hamming code) для корекції помилок. Потік даних розбивається на слова – причому розмір слова відповідає кількості дисків для запису даних. Для кожного слова обчислюється код корекції помилок, що записується на диски, виділені для зберігання контрольної інформації. Їхнє число дорівнює кількості біт у слові контрольної суми.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

RAID 2 не одержав комерційного застосування.

Якщо слово складається із чотирьох біт, то під контрольну інформацію приділяється три диски. RAID 2 – один з деяких рівнів, що дозволяють виявляти подвійні помилки й виправляти "на льоту" одиночні. При цьому він є самим надлишковим серед всіх рівнів з контролем парності. Ця схема зберігання даних не одержала комерційного застосування, оскільки погано справляється з великою кількістю запитів.

Переваги:

- досить проста реалізація;
- корекція помилок "на льоту";
- дуже висока швидкість передачі даних;
- при збільшенні кількості дисків накладні витрати зменшуються.

Недоліки:

- низька швидкість обробки запитів;
- висока вартість;
- більша надмірність.

RAID 3 – Відказостійкий дисковий масив з паралельною передачею даних і парністю (Parallel Transfer Disks with Parity)

Відказостійкий масив з паралельним вводом/виводом даних і диском контролю парності. Потік даних розбивається на порції на рівні байт (хоча можливо й на рівні біт) і записується одночасно на всі диски масиву, крім одного. Один диск призначений для зберігання контрольних сум, що обчислюються при записі даних. Поломка кожного з дисків масиву не приведе до втрати інформації.

В RAID 3 інформація розбивається на порції однакового розміру.

Цей рівень має набагато меншу надмірність, чим RAID 2. У RAID 2 більшість дисків, що зберігають контрольну інформацію, потрібні для визначення несправного розряду. Як правило, RAID-контролери можуть одержати дані про помилку за допомогою механізмів відстеження випадкових збоїв. За рахунок розбивки даних на порції RAID 3 має високу продуктивність. Оскільки при

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

кожній операції вводу/виводу виробляється обіг практично до всіх дисків масиву, те одночасна обробка декількох запитів неможлива.

Цей рівень підходить для додатків з файлами великого обсягу й малою частотою звертань (в основному це сфера мультимедіа). Використання тільки одного диска для зберігання контрольної інформації пояснює той факт, що коефіцієнт використання дискового простору досить високий (як наслідок цього – відносно низька вартість). Для реалізації масиву потрібно не менше трьох вінчестерів.

Переваги:

- відмова диска мало впливає на швидкість роботи масиву;
- висока швидкість передачі даних;
- високий коефіцієнт використання дискового простору.

Недоліки:

- складність реалізації;
- низька продуктивність при великій інтенсивності запитів даних невеликого обсягу.

RAID 4 – Відказостійкий масив незалежних дисків із загальним диском парності (Independent Data Disks with Shared Parity Disk)

Цей масив дуже схожий на рівень RAID 3. Потік даних розділяється не на рівні байтів, а на рівні блоків інформації, кожний з яких записується на окремий диск. Після запису групи блоків обчислюється контрольна сума, що записується на виділений для цього диск.

В RAID 4 потік даних розділяється на блоки.

В RAID 4 можливо одночасне виконання декількох операцій читання. Цей масив підвищує продуктивність передачі файлів малого обсягу (за рахунок розпаралелювання операції зчитування). Але оскільки при записі повинна змінюватися контрольна сума на виділеному диску, одночасне виконання операцій неможливо (у наявності асиметричність операцій вводу й виводу). Цей рівень має майже всі недоліки RAID 3 і не забезпечує переваги у швидкості при

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

передачі даних великого обсягу. Схема зберігання розроблялася для додатків, у яких дані споконвічно розбиті на невеликі блоки, тому немає необхідності розбивати їх додатково. Ця схема зберігання даних має невисоку вартість, але її реалізація досить складна, як і відновлення даних при збої.

Переваги:

- висока швидкість передачі даних;
- відмова диска мало впливає на швидкість роботи масиву;
- високий коефіцієнт використання дискового простору.

Недоліки:

- досить складна реалізація;
- дуже низька продуктивність при записі даних;
- складне відновлення даних.

RAID 5 – Відказостійкий масив незалежних дисків з розподіленою парністю (Independent Data Disks with Distributed Parity Blocks)

Найпоширеніший рівень. Блоки даних і контрольних сум циклічно записуються на всі диски масиву, відсутній виділений диск для зберігання інформації про парність, немає асиметричності конфігурації дисків.

У випадку RAID 5 всі диски масиву мають однаковий розмір – але один з них не бачимий для операційної системи. Наприклад, якщо масив складається з п'яти дисків ємністю 10 Гб кожний, те фактично розмір масиву буде дорівнює 40 Гб – 10 Гб приділяється на контрольні суми. У загальному випадку корисна ємність масиву з N дисків дорівнює сумарної ємності N- 1 диска.

В RAID 5 відсутній виділений диск для зберігання інформації про парність.

Найбільший недолік рівнів RAID від 2-го до 4-го – це наявність окремого диска (або дисків), що зберігає інформацію про парність. Швидкість виконання операцій зчитування досить висока, тому що не вимагає звертання до цього диска. Але при кожній операції запису на ньому змінюється інформація, тому схеми RAID 2-4 не дозволяють проводити паралельні операції запису. RAID 5 не

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

має цього недоліку, тому що контрольні суми записуються на всі диски масиву, що уможлиблює виконання декількох операцій читання або записи одночасно. RAID 5 має досить високу швидкість запису/читання й малу надмірність.

Переваги:

- висока швидкість запису даних;
- досить висока швидкість читання даних;
- висока продуктивність при великій інтенсивності запитів читання/запису даних;
- високий коефіцієнт використання дискового простору.

Недоліки:

- низька швидкість читання/запису даних малого обсягу при одиничних запитах;
- досить складна реалізація;
- складне відновлення даних.

RAID 6 – Відказостійкий масив незалежних дисків із двома незалежними розподіленими схемами парності (Independent Data Disks with Two Independent Distributed Parity Schemes)

RAID 6 – це відказостійкий масив незалежних дисків з розподілом контрольних сум, обчислених двома незалежними способами. Цей рівень багато в чому схожий з RAID 5. Тільки в ньому використовується не одна, а дві незалежні схеми контролю парності, що дозволяє зберігати працездатність системи при одночасному виході з ладу двох накопичувачів. Для обчислення контрольних сум в RAID 6 використовується алгоритм, побудований на основі коду Ріда-Соломона (Reed-Solomon).

RAID 6 використовує дві незалежні схеми контролю парності.

Цей рівень має дуже високу відказостійкість, більшу швидкість зчитування (дані зберігаються блоками, немає виділених дисків для зберігання контрольних сум). У той же час через великий обсяг контрольної інформації RAID 6 має низьку швидкість запису. Він дуже складний у реалізації,

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

характеризується низьким коефіцієнтом використання дискового простору: для масиву з п'яти дисків він становить усього 60%, але з ростом числа дисків ситуація виправляється.

RAID 6 по багатьом характеристикам програє іншим рівням, тому на сьогодні не одержав комерційного застосування.

Переваги:

- висока відказостійкість;
- досить висока швидкість обробки запитів;

Недоліки:

- низька швидкість читання/запису даних малого обсягу при одиничних запитах;
- дуже складна реалізація;
- складне відновлення даних;
- низька швидкість запису даних.

RAID 7 – Відказостійкий масив, оптимізований для підвищення продуктивності (Optimized Asynchrony for High I/O Rates as well as High Data Transfer Rates)

На відміну від інших рівнів, RAID 7 не є відкритим індустріальним стандартом – це зареєстрована торговельна марка компанії Storage Computer Corporation. Масив ґрунтується на концепціях, використаних у третьому й четвертому рівнях. Додалася можливість кешування даних. До складу RAID 7 входить контролер з убудованим мікропроцесором під керуванням операційної системи реального часу (real-time OS). Вона дозволяє обробляти всі запити на передачу даних асинхронно й незалежно.

RAID 7 – зареєстрована торговельна марка компанії Storage Computer Corporation.

Блок обчислення контрольних сум інтегрований із блоком буферізації; для зберігання інформації про парність використовується окремий диск, що може бути розміщений на будь-якому каналі. RAID 7 має високу швидкість передачі

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

даних і обробки запитів, гарну масштабованість. Самим більшим недоліком цього рівня є вартість його реалізації.

Переваги:

- дуже висока швидкість передачі даних і висока швидкість обробки запитів (в 1,5...6 разів вище інших стандартних рівнів RAID);
- гарна масштабованість;
- значно зросла (завдяки наявності кешу) швидкість читання даних невеликого обсягу;
- відсутність необхідності в додатковій передачі даних для обчислення парності.

Недоліки:

- власність однієї компанії;
- складність реалізації;
- дуже висока вартість на одиниці об'єму;
- не може обслуговуватися користувачем;
- необхідність використання блоку безперебійного живлення для запобігання втрати даних з кеш-пам'яті;
- короткий гарантійний строк.

RAID 10

RAID 10 – відказостійкий масив з дублюванням і паралельною обробкою. Ця архітектура виявляє собою масив типу RAID 0, сегментами якого є масиви RAID 1. Він поєднує в собі високу відказостійкість і продуктивність.

Комбіновані рівні

Крім базових рівнів RAID 0 – RAID 5, описаних у стандарті, існують комбіновані рівні RAID 1+0, RAID 3+0, RAID 5+0, RAID 1+5, які різні виробники інтерпретують кожний по-своєму.

- RAID 1+0 – це сполучення дзеркалювання й чергування.

Нинішні контролери використовують цей режим за замовчуванням для RAID 1. Тобто, 1 диск основний, 2-й диск – дзеркало, причому читання

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

виробляється з них по черзі, як для RAID 0. Властиво, зараз можна вважати що RAID 1 і RAID 1+0 – це просто різна назва того самого методу апаратного дзеркалювання дисків. Але не варто забувати, що повноцінний RAID 1+0 повинен містити як мінімум 4 диски.

– RAID 5+0 – це чергування томів 5-го рівня. RAID 1+5 – RAID 5 із дзеркальованою парою.

Комбіновані рівні успадковують як переваги, так і недоліки своїх «батьків»: поява чергування в рівні RAID 5+0 анітрошки не додає йому надійності, але зате позитивно відбивається на продуктивності. Рівень RAID 1+5, напевно, дуже надійний, але не найшвидший і, до того ж, у край неекономічний: корисна ємність тому менше половини сумарної ємності дисків.

Варто відзначити, що кількість жорстких дисків у комбінованих масивах також зміниться. Наприклад для RAID 5+0 використовують 6 або 8 жорстких дисків, для RAID 1+0 – 4, 6 або 8.

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Система складається з наступних блоків.

Блок вибору кількості дисків – призначений для визначення кількості дисків з яких буде складатися RAID-масив. Якщо дисків буде 2 то буде використовуватися технологія RAID 0 та RAID 1. Якщо дисків буде більше 2 то буде використовуватися технологія RAID 0 та RAID 6.

Блок формування віртуальних дисків використовується у разі коли диск тільки один. Тоді на цьому диску формуються декілька віртуальних дисків, з якими відбуваються дії аналогічні як ті, що відбуваються над фізичними дисками. Тобто RAID-масив формується з віртуальних дисків.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

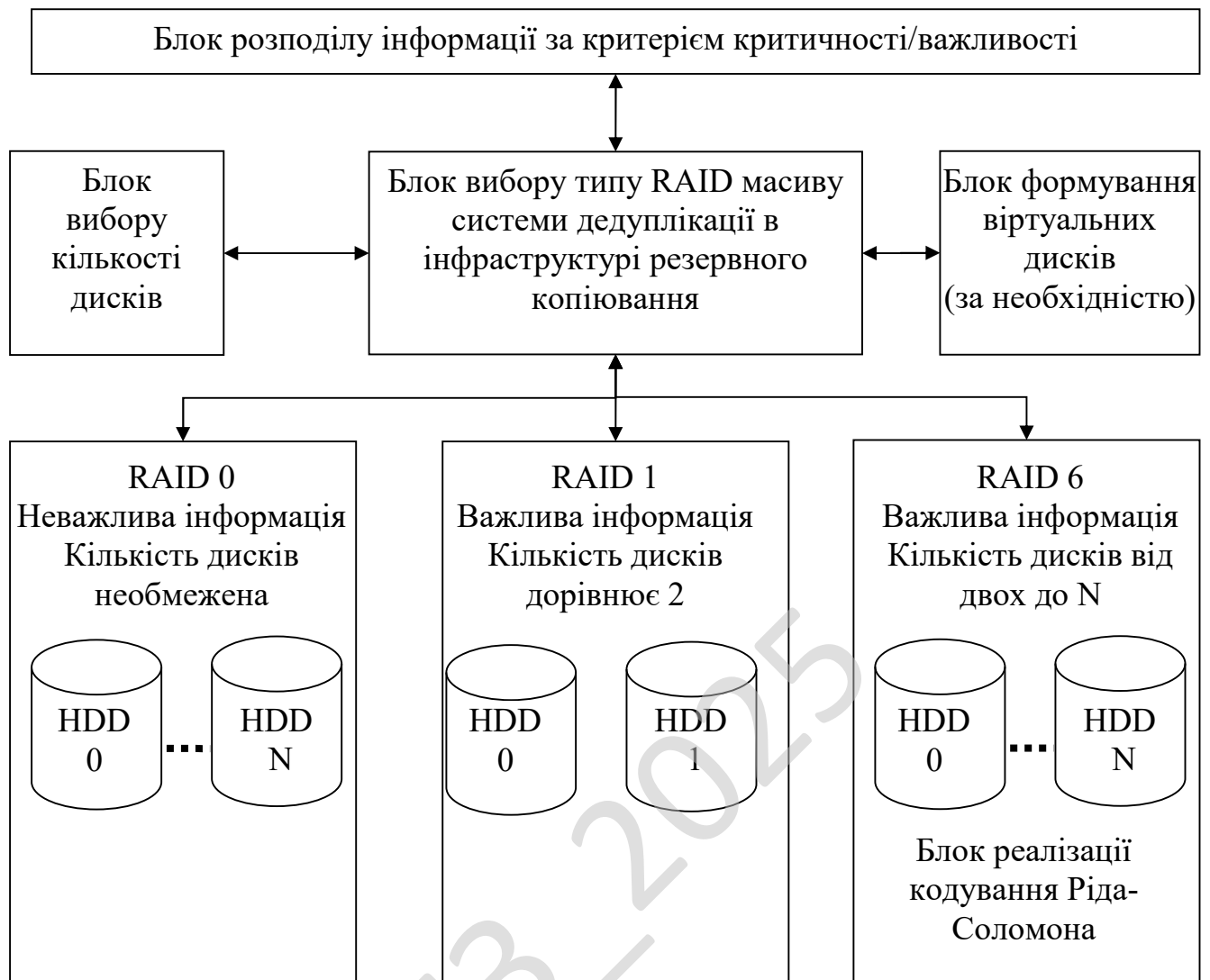


Рисунок 3.2 – Функціональна схема системи

Блок розподілу інформації за критерієм критичності/важливості визначає який саме вид RAID-масиву необхідно буде використовувати. Якщо інформація не є критичною то буде використовуватися RAID 0. Якщо інформація є важливою, то в залежності від ступеня важливості, та вимог до швидкодії, буде використовуватися або RAID 1, який є більш швидким, та менш надійним, або RAID 6, який є дуже надійним але менш швидким.

Блок вибору типу RAID масиву дозволяє автоматично, в залежності від заданих параметрів у попередніх блоках, визначити, який саме тип RAID-масиву потрібен для зберігання тієї, або іншої інформації.

$g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0 + 1}) \dots (x - \alpha^{l_0 + d - 1})$; 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 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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

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 поліном, 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 $l \ln n^2$.

Кодування

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

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

– 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 $GF(0)...GF(15)$ 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 кодера 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.

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.

На структурній схемі кодера (рисунок 3.2) символу «С» відповідають дві константи $GF(n)$.

						ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			66

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 (на рисунку не показана) 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 досяглася 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.

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

3.4 Розробка діаграми процесів

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників.

Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

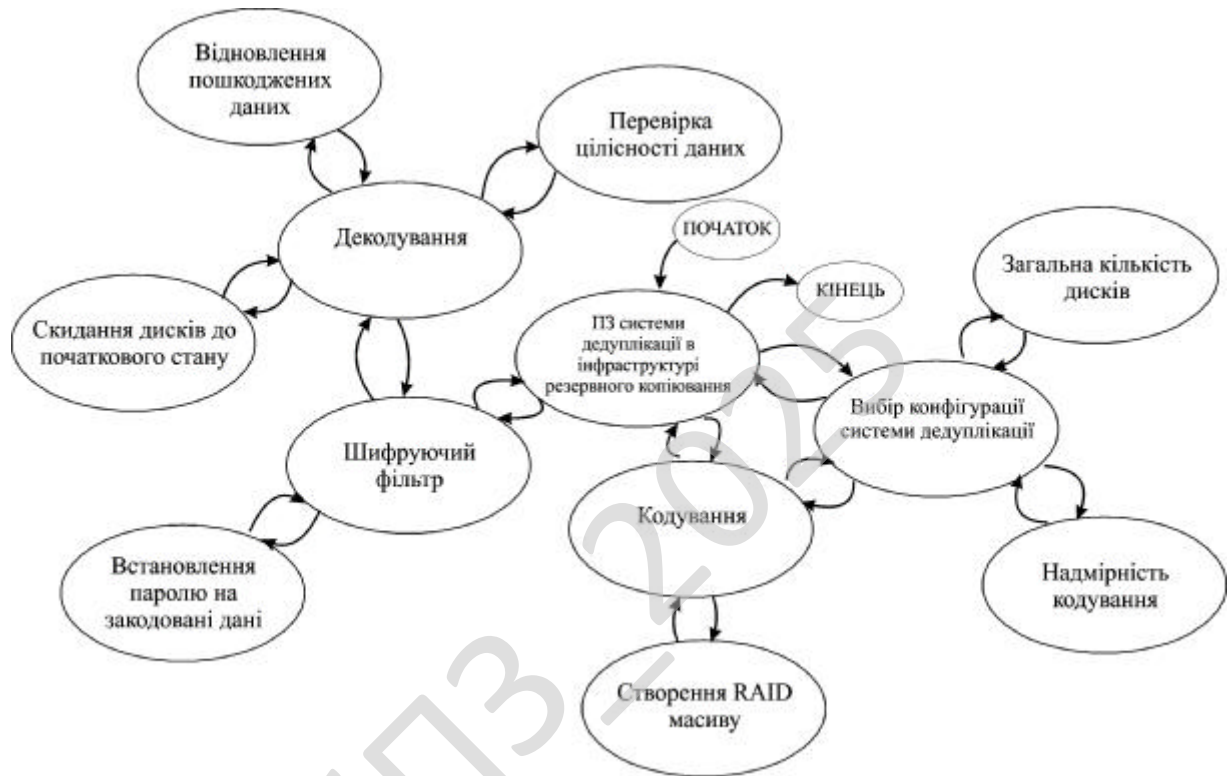


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

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

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

Діаграми потоків даних містять чотири типи елементів:

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

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.

КБПЗ – 2025

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

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

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

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

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

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

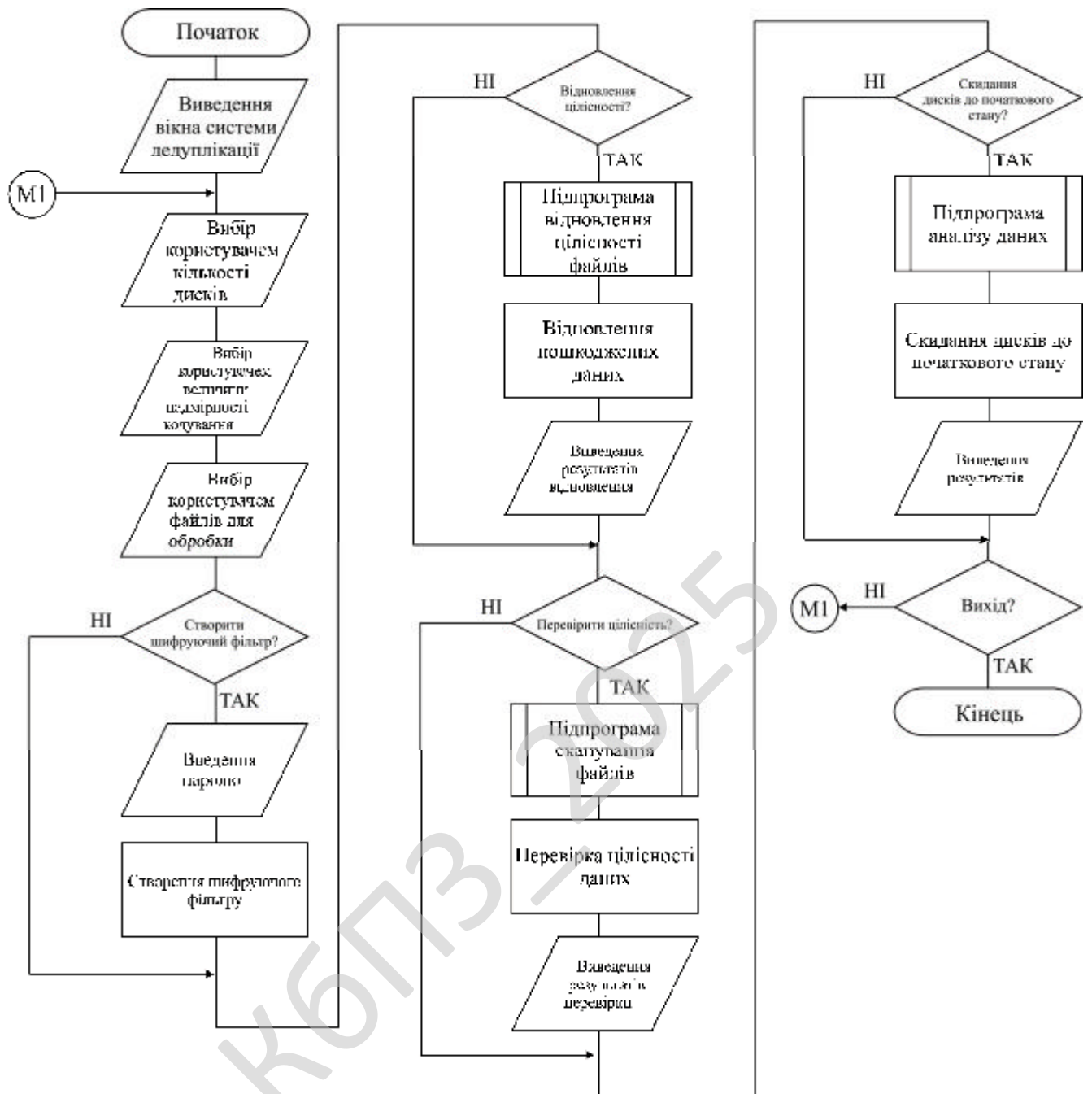


Рисунок 4.1 – Блок-схема основної програми

Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

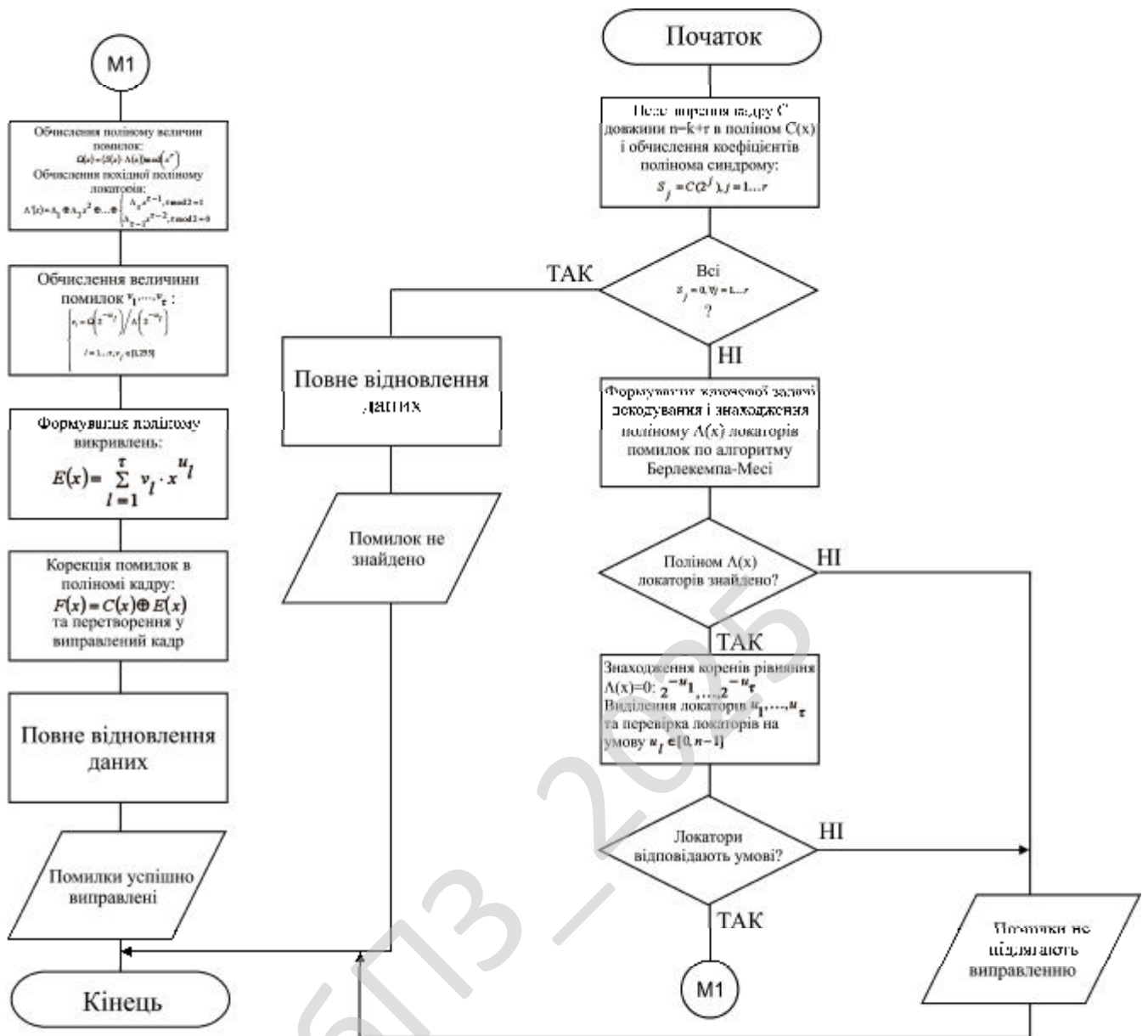


Рисунок 4.2 – Блок-схема роботи підпрограми

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine

написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.

– Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Vazaar и Darcs).

- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами. Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

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

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

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

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

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

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

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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

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

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

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

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

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

- Виправлено (виправлення включені у версію таку-то).
- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).
- «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

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

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

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

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

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ІС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

– Німецька V-модель була розроблена аерокосмічною компанією IAVG в Оттобрунні поряд з Мюнхеном у сприянні з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини. Модель була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року. V-модель використовується для управління процесом розробки програмного забезпечення для німецької федеральної адміністрації.

Зараз вона є стандартом для німецьких урядових і оборонних проєктів, а також для виробників ПЗ в Німеччині. V-Model являє собою скоріше набір стандартів у галузі проєктів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проєктного управління, так і для системного розвитку.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проєкту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проєкті виробляються по горизонталі, між лівою і правою сторонами літери.

Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. У середині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування.

Модель базується на тому, що приймально-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектури, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти відхилення в проекті і ризики на ранніх стадіях і покращує якість управління проектом.

2. Підвищення та гарантії якості: V-Model – стандартизована модель розробки, що дозволяє домогтися від проекту результатів бажаної якості. Проміжні результати можуть бути перевірені на ранніх стадіях. Універсальне документування полегшує читаність, зрозумілість та контрольованість.

3. Зменшення загальної вартості проекту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проекти.

4. Підвищення якості комунікації між учасниками проекту: Універсальний опис усіх елементів та умов полегшує взаєморозуміння всіх учасників проекту. Таким чином, зменшуються неточності у розумінні між користувачем, покупцем, постачальником і розробником.

Переваги:

– Користувачі V-Model беруть участь у розробці та підтримці V-моделі. Комітет з контролю за змінами підтримує проект і збирається раз на рік для обробки всіх отриманих запитів на внесення змін до V-Model.

– На старті будь-якого проекту V-подібна модель може бути адаптована під цей проект, так як ця модель не залежить від типів організацій та проектів.

– V-model дозволяє розбити діяльність на окремі кроки, кожен з яких буде включати в себе необхідні для нього дії, інструкції до них, рекомендації та докладне пояснення діяльності.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Обмеження. Наступні моменти не враховуються в V-моделі, але можуть бути розглянуті окремо, або можливо адаптувати модель під них:

– Не регулюється розміщення контрактів на обслуговування.

– Організація і виконання управління, обслуговування, ремонту та утилізації системи не враховуються в V-моделі. Однак, планування і підготовка до цих операцій моделлю розглядаються.

– V-подібна модель більше стосується розробки програмного забезпечення в проекті, ніж всієї організації процесу.

Переваги:

– У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.

– У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.

– У V-подібній моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПО – перед розробкою компонентів.

– Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожен отриманий дані повинні піддаватися тестуванню.

– Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

Недоліки:

– Модель не передбачає роботу з паралельними подіями.

– У моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

– Тестування вимог в життєвому циклі відбувається занадто пізно, внаслідок чого неможливо внести змін, не вплинувши при цьому на графік виконання проекту.

– У модель не входять дії, спрямовані на аналіз ризиків.

– Деякий результат можна отримати тільки при досягненні низу букви V.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

– набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;

– набір клієнтів, які використовують сервіси, що надаються серверами;

– мережа, яка забезпечує взаємодію між клієнтами та серверами.

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

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

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

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

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

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

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

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

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

підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

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

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

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

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

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

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

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

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

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

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою MARS, який є блочно-симетричним шифром з відкритим ключем. Розмір блоку при шифруванні 128 біта, розмір ключа може варіюватися від 128 до 448 біт включно (кратні 32бітам). Творці прагнули поєднати в своєму алгоритмі швидкість кодування і стійкість шифру. В результаті вийшов один з самих криптостійкий алгоритм з алгоритмів, які брали участь в конкурсі AES.

Алгоритм унікальний тим, що використовував практично всі існуючі технології, застосовувані в криптоалгоритмах, а саме:

- Найпростіші операції (додавання, віднімання, виключаюче або).
- Підстановки з використанням таблиці замін.
- Фіксований циклічний зсув.
- Залежний від даних циклічний зсув.
- Множення за модулем 2^{32} .
- Ключове забілювання.

Використання подвійного перемішування представляє складність для криптоаналізу, що деякі відносять до недоліків алгоритму. У той же час на даний момент не існує будь-яких ефективних атак на алгоритм, хоча деякі ключі можуть генерувати слабкі підключі.

Структура алгоритму

Автори шифру виходили з наступних припущень:

1. Вибір операцій. MARS був спроектований для використання на найсучасніших комп'ютерах того часу. Для досягнення найкращих захисних характеристик в нього були включені самі «сильні операції» підтримувані в них. Це дозволило добитися більшого відношення securityper-instruction для різних реалізацій шифру.

2. Структура шифру. Двадцятирічний досвід роботи в області криптографії підштовхнув творців алгоритму до думки, що кожен раунд

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

шифрування грає свою роль в забезпеченні безпеки шифру. Зокрема, ми можемо бачити, що перший і останній раунди зазвичай сильно відрізняються від проміжних («центральных») раундів алгоритму в плані захисту від криптоаналітичних атак. Таким чином, при створенні MARSa використовувалася змішана структура, де перший і останній раунди шифрування істотно відрізняються від проміжних.

3. Аналіз. Швидше за все, алгоритм з гетерогенною структурою буде краще протистояти криптоаналітичним методам майбутнього, ніж алгоритм, всі раунди якого ідентичні. Розробники алгоритму MARS надали йому сильно гетерогенну структуру – раунди алгоритму дуже різняться між собою.

У шифрі MARS використовувалися такі методи шифрування:

1. Робота з 32-х бітними словами. Всі операції застосовуються до 32-бітовим словами. тобто вся початкова інформація розбивається на блоки по 32біта. (Якщо ж блок опинявся меншої довжини, то він доповнювався до 32біт)

2. Мережа Фейстеля. Творці шифру вважали, що це найкращий варіант поєднання швидкості шифрування і криптостійкості. В MARS використана мережа Фейстеля 3-го типу.

3. Симетричність алгоритму. Для стійкості шифру до різних атакам всі його раунди були зроблені повністю симетричними, тобто друга частина раунду є дзеркальне повторення першої його частини.

Структуру алгоритму MARS можна описати таким чином:

1. Попереднє накладення ключа: на 32-бітові субблоки A, B, C, D накладаються 4 фрагмента розширеного ключа $k_0 \dots k_3$ операцією складання за модулем 2^{32} .

2. Виконуються 8 раундів прямого перемішування (без участі ключа шифрування).

3. Виконуються 8 раундів прямого криптоперетворення.

4. Виконуються 8 раундів зворотного криптоперетворення. [2]

5. Виконуються 8 раундів зворотного перемішування, також без участі ключа шифрування.

6. Фінальне накладення фрагментів розширеного ключа $k_{36} \dots k_{39}$ операцією віднімання за модулем 2^{32} .

Пряме перемішування

У першій фазі на кожне слово даних накладається слово ключа, а потім відбувається вісім раундів змішування згідно з мережею Фейстеля третього типу спільно з деякими додатковими змішування. У кожному раунді ми використовуємо одне слово даних (зване, вихідним словом) для модифікації трьох інших слів (звані, цільовими словами). Ми розглядаємо чотири байта вихідного слова як індексів на двох S-блоків, S_0 і S_1 , кожен, що складається з 256 32-розрядних слів, а далі проводимо операції XOR або додавання даних відповідного S-блоку в три інших слова.

Якщо чотири байти вихідного слова b_0, b_1, b_2, b_3 (де b_0 є першим байтом, а b_3 є старшим байтом), то ми використовуємо b_0, b_2 , як індекси в блоку S_0 і байти b_1, b_3 , як індекси в S-блоці S_1 . Спочатку зробимо XOR S_0 до першого цільовим речі, а потім додамо S_1 до того ж слова. Ми також додаємо S_0 до другого цільовим слову і XOR блоку- S_1 до третього цільовим слову. У висновку, ми обертаємо вихідне слово на 24 біта вправо.

У наступному раунді ми обертаємо наявні у нас чотири слова: таким чином, нинішні перші цільове слово стає наступним вихідним словом, поточним другий цільове слово стає новим першим цільовим словом, третє цільове слово стає наступний другий цільовим словом, і поточне вихідне слово стає третім цільовим словом.

Більш того, після кожного з чотирьох раундів ми додаємо одне з цільових слів назад у вихідне слово. Зокрема, після першого і п'ятого раундів ми додав третю цільове слово назад у вихідне слово, а після другого і шостого раунду ми додаємо першої цільової слово назад у вихідне слово. Причиною цих додаткових операцій змішування, є ліквідація декількох простих диференціальних

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

криптоатаки в фазі перемішування, щоб порушити симетрію у фазі змішування та отримати швидкий потік.

Криптографічне ядро

Криптографічне ядро MARS – мережа Фейстеля 3-го типу, що містить в собі 16 раундів. У кожному раунді ми використовуємо ключову E-функцію, яка є комбінацією множень, обертань, а також звернень до S-блоків. Функція приймає на вхід слово даних, а повертає три слова, з якими згодом буде здійснена операція додавання або XOR до інших трьох слів даними. У доповненні вихідне слово обертається на 13 біт вліво.

Для забезпечення, серйозного опору до криптоатаки, три вихідних значення E-функції (O_1 , O_2 , O_3) використовуються в перших восьми раундах і в останніх восьми раундах в різних порядках. У перші вісім раундів ми додаємо O_1 і O_2 до першого і другого цільовим речі, відповідно, і XOR O_3 у третьому цільовим слову. За останні вісім раундів, ми додаємо O_1 і O_2 до третього і другого цільовим речі, відповідно, і XOR O_3 до першого цільовим слову.

E-функція

E-функція приймає як вхідні дані одне слово даних і використовує ще два ключових слова, виробляючи на виході три слова. У цій функції ми використовуємо три тимчасові змінні, що позначаються L, M і R (для лівої, середньої та правої).

Спочатку ми встановлюємо в R значення вихідного слова зміщеного на 13 біт вліво, а в M – сума вихідних слів і першого ключового слова. Потім ми використовуємо перші дев'ять бітів M як індекс до однієї з 512S-блоків (яке виходить суміщенням S_0 і S_1 змішуванням фази), і зберігаємо в L значення відповідного S-блоку.

Потім помножимо друге ключове слово на R, зберігши значення в R. Потім обертаємо R на 5 позицій вліво (так, 5 старших бітів стають 5 нижніми бітами R після обертання). Тоді ми робимо XOR R в L, а також переглядаємо п'ять нижніх біт R для визначення величини зсуву (від 0 до 31), і обертаємо M

вліво на цю величину. Далі ми обертаємо R ще на 5 позицій вліво і робимо XOR в L. У висновку, ми знову дивимося на 5 молодших бітів R, як на величину обертання і обертаємо L на цю величину вліво. Таким чином результат роботи E-функції – 3 слова (по порядку): L, M, R.

Зворотне перемішування

Зворотне перемішування практично збігається з прямим перемішуванням, за винятком того факту, що дані обробляються в зворотному порядку. Тобто, якби ми поєднали пряме і зворотне перемішування так, щоб їх виходи і входи були б з'єднані в зворотному порядку ($D[0]$ прямого і $D[3]$ зворотного, $D[1]$ прямого і $D[2]$ зворотного), то не побачили б результату перемішування. Як і в прямому змішуванні, тут ми теж використовуємо одне вихідне слово і три цільових. Розглянемо чотири перших байта вихідного слова: b_0, b_1, b_2, b_3 . Будемо використовувати b_0, b_2 як індекс до S-блоку – S_1 , а b_1, b_3 для S_0 . Зробимо XOR $S_1[b_0]$ в перше цільове слово, віднімемо $S_0[b_3]$ з другого слова, віднімемо $S_1[b_2]$ з третього цільового слів і потім проробимо XOR $S_0[b_1]$ також до третього цільового слова. Нарешті, ми повертаємо початкове слово на 24 позицій вліво. Для наступного раунду ми обертаємо наявні слова так, щоб нинішнє перше цільове слово стало наступним вихідним словом, поточне друге цільове слово стало першим цільовим словом, поточне третє цільове слово стало другим цільовим словом, і поточне вихідне слово стало третім цільовим словом. Крім того, перед одним з чотирьох «особливих» раундів ми віднімаємо одне з цільових слів з вихідного слова: перед четвертим і восьмим раундами ми віднімаємо першої цільової слово, перед третьому і сьомим раундами ми віднімемо третя цільова слово з вихідного.

Дешифрування

Процес декодування обернений процесу кодування. Код дешифрування схожий (але не ідентичний) на код шифрування.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку зображено розроблене у магістерської дипломної роботі ПЗ системи дедуплікації в інфраструктурі резервного копіювання. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Розділу обрання групи.
- Розділу виведення результату роботи системи.
- Функції представлені у графічному вигляді.

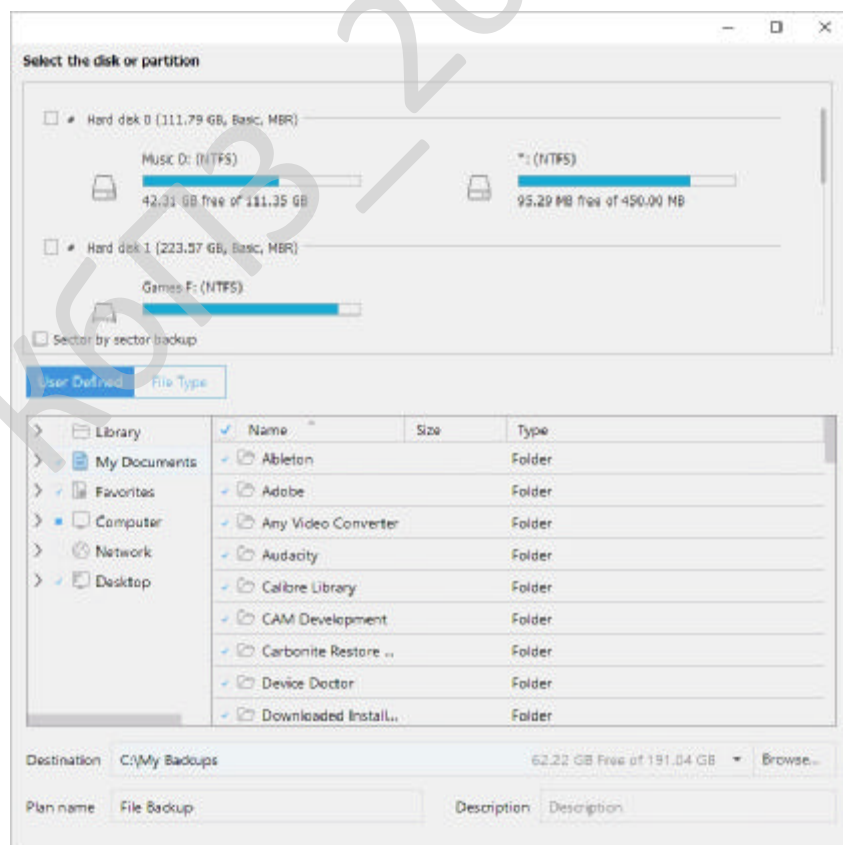


Рисунок 5.1 – Головне вікно ПЗ

Дедуплікацію даних для стислості часто називають Dedup. Це функція, за допомогою якої можливо зменшити вплив надлишкових даних на вартість зберігання. Якщо дедуплікація даних включена, вона оптимізує вільне місце в тому ж за рахунок перевірки дані томи на наявність частин, що дублюються. частини, Що Дублюються, набору дані томи зберігаються один раз і (при необхідності) стискаються для додаткової економії. Дедуплікація оптимізує надлишкові дані, не порушуючи вірогідність або цілісність даних.

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

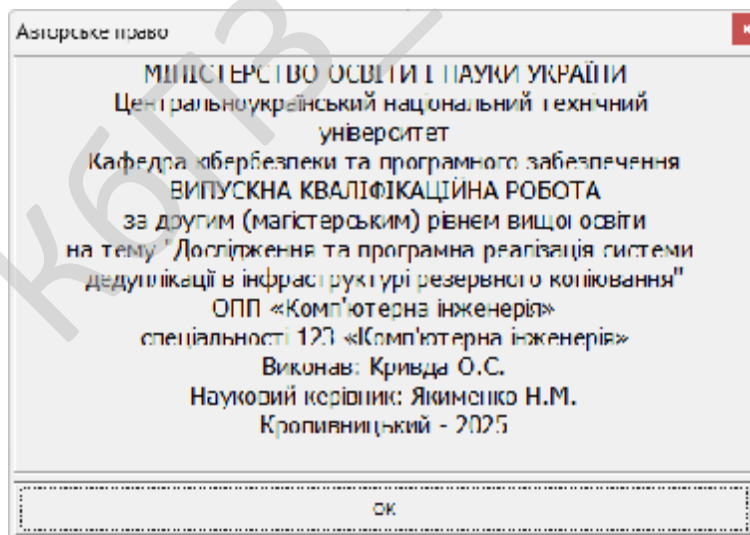


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

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

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

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

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

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

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

Проводилась оцінка:

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

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

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

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

Ці тести демонструють:

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

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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 – commercial software. Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційних програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації, надають відкриті вихідні коди. Для продуктів, що розповсюджуються на комерційній основі діють зазвичай безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проєктів. Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям. Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм.

Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють

вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником.

Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

КБПЗ_2025

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи дедуплікації в інфраструктурі резервного копіювання.

Метою розробки є дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання.

Об'єктом дослідження є процес дедуплікації в інфраструктурі резервного копіювання.

Предметом дослідження є методи дедуплікації в інфраструктурі резервного копіювання.

Методи дослідження базуються на методах 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 дедуплікації в інфраструктурі резервного копіювання.

– Розроблено вітчизняний продукт дедуплікації в інфраструктурі резервного копіювання, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження і практичної реалізації системи дедуплікації будуть особливо цікавими для компаній, які мають справу з великими обсягами даних і активно використовують резервне копіювання. Це можуть бути фінансові установи, телекомунікаційні компанії, державні структури, медичні організації, а також підприємства з розвиненою ІТ-інфраструктурою. Для таких організацій питання економії простору зберігання і підвищення ефективності копіювання є надзвичайно актуальним.

Також система дедуплікації буде корисною для дата-центрів і хмарних провайдерів, які надають послуги зберігання та резервування даних клієнтів. Завдяки впровадженню цієї технології вони можуть суттєво скоротити обсяги необхідного сховища, а отже, знизити експлуатаційні витрати й підвищити рентабельність своїх послуг.

Крім того, результати дослідження можуть бути цінними для ІТ-департаментів навчальних закладів, наукових центрів або великих підприємств, які зберігають архіви проєктів, звітів, баз даних чи відеоматеріалів. Усі ці установи потребують ефективних рішень для роботи з великими масивами інформації, і система дедуплікації допомагає вирішити проблему надлишкового дублювання даних.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для визначення привабливості впровадження системи дедуплікації доцільно скористатися методом експертних оцінок. До групи експертів можна залучити фахівців у галузі інформаційних технологій, адміністраторів дата-центрів, керівників ІТ-департаментів та представників бізнесу, що мають практичний досвід зберігання великих обсягів даних. Вони оцінюють проєкт за критеріями – очікуваний рівень економії ресурсів, технічна складність, масштабованість, сумісність із наявною інфраструктурою та термін окупності.

Наприклад, за підсумками експертного опитування, система може отримати 9 балів за економічну ефективність, 8 за технічну реалістичність, 7 за масштабованість і 9 за рівень корисності. Середній бал 8,25 із 10 підтвердить, що проєкт має високий потенціал і є доцільним для впровадження.

Такі результати експертного оцінювання допомагають сформуванню обґрунтованого рішення для керівництва підприємства щодо фінансування проєкту. Крім того, вони дозволяють виділити сильні сторони розробки та заздалегідь визначити можливі ризики, що підвищує якість управлінських рішень на етапі реалізації.

7.3 Вибір методу оцінки вартості ПЗ

Оптимальним методом оцінки вартості впровадження системи дедуплікації є метод повної вартості володіння (TCO – Total Cost of Ownership). Він дозволяє оцінити всі прямі та непрямі витрати, пов'язані з придбанням, розгортанням, налаштуванням та обслуговуванням системи. До таких витрат належать вартість обладнання, витрати на навчання персоналу, підтримку системи, а також енергоспоживання та обслуговування серверів.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

Цей метод допомагає не лише визначити, скільки підприємство витратить на впровадження, але й показує, які саме витрати можна оптимізувати у майбутньому. Наприклад, зменшення потреби в дисковому просторі й скорочення часу на створення резервних копій одразу впливають на зниження операційних витрат.

Поєднання TCO з аналізом вигід (Cost-Benefit Analysis) дозволяє наочно порівняти вкладені ресурси з отриманими результатами, такими як зменшення витрат на сховище чи підвищення продуктивності роботи IT-персоналу. Це робить аналіз вартості більш прозорим і зрозумілим для управлінців, які ухвалюють рішення про інвестиції у технологію.

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

Економічна ефективність від впровадження системи дедуплікації в інфраструктурі резервного копіювання на прикладі середнього підприємства з обсягом даних близько 100 ТБ, які регулярно архівуються для збереження безпеки та відновлення у разі збою. Вхідні дані зафіксовано в таблиці 7.1.

Розрахунок економічного ефекту демонструє наступне: економія на зберіганні даних – 720 000грн, зменшення витрат на обслуговування – 60 000грн, економія енергії та охолодження – 144 000грн, підвищення ефективності персоналу (менше часу на резервування та відновлення, еквівалент 0,25 ставки адміністратора) – 45 000грн, загальний річний ефект – 969 000 грн, річні витрати на підтримку системи – 50 000 грн, чистий річний економічний ефект – 919 000 грн, термін окупності – 0,33 року (~4 місяці), ROI (окупність інвестицій) – 306%.

Додаткові нефінансові ефекти: підвищення швидкості резервного копіювання на 40–60%, що зменшує простої у роботі серверів, зниження ризику втрати даних завдяки централізованому управлінню резервами і зменшенню

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

кількості дубльованих копій, покращення контролю за IT-інфраструктурою – адміністратори отримують інструменти для моніторингу, аналітики та оптимізації політик резервування, можливість масштабування – при зростанні обсягів даних не потрібно одразу купувати нове обладнання, відповідність стандартам безпеки – зменшення навантаження на мережеві канали під час копіювання знижує ризик втрати або пошкодження інформації.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження
Загальний обсяг резервних копій	100 ТБ	25 ТБ
Кількість копій одних і тих самих файлів	до 60%	<5%
Витрати на зберігання (1 ТБ/міс)	800 грн	800 грн
Річні витрати на сховище (без дедуплікації)	960 000 грн	—
Витрати на обслуговування (енергія, персонал)	120 000 грн	60 000 грн
Початкові інвестиції у систему дедуплікації	—	300 000 грн
Річні витрати на підтримку системи	—	50 000 грн

Окрім фінансових вигод, підприємство отримує стабільну, безпечну та енергоефективну систему резервного збереження даних, що відповідає вимогам сучасних IT-інфраструктур. У довгостроковій перспективі це рішення не лише оптимізує витрати, а й підвищує стійкість бізнесу до технічних і кіберризиків.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування проєкту системи дедуплікації має ґрунтуватися на демонстрації її практичних переваг та економічної вигоди. На початковому етапі варто створити пілотну версію або демонстраційну інсталяцію, щоб потенційні клієнти могли побачити, як система працює з реальними даними. Показ практичних результатів – найефективніший спосіб переконати компанії у доцільності інвестицій у цей продукт.

Після цього можна проводити інформаційні кампанії на IT-конференціях, вебінарах і профільних форумах. Публікації у фахових журналах і створення коротких відео-презентацій із реальними кейсами також підвищують довіру до розробки. Такий підхід допомагає донести інформацію до цільової аудиторії – керівників IT-департаментів, адміністраторів систем зберігання даних і менеджерів IT-компаній.

На фінальному етапі важливо налагодити партнерські зв'язки з виробниками апаратного забезпечення, постачальниками серверів та сховищ. Спільні маркетингові програми і пропозиції «під ключ» створюють ефект синергії та збільшують охоплення ринку.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для ефективної реалізації проєкту слід комбінувати прямі та партнерські канали збуту. Один із напрямів – співпраця з IT-інтеграторами, які вже мають клієнтську базу серед великих підприємств. Вони можуть включати систему дедуплікації у свої готові рішення для корпоративного резервного копіювання.

Інший варіант – реалізація системи за моделлю SaaS (Software-as-a-Service), коли користувачі отримують доступ до функціоналу через хмару без необхідності купівлі обладнання. Це відкриває проєкту шлях до малого та середнього бізнесу, який не має власних серверних потужностей.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

Також важливо розвивати власні маркетингові канали: сайт із технічною документацією, кейсами успішних впроваджень і калькулятором економічної ефективності. Це не лише підвищує рівень довіри до продукту, а й дозволяє потенційному клієнту самостійно оцінити вигоди від впровадження.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключовими факторами успіху є стабільність роботи системи, її масштабованість і простота інтеграції з уже наявними рішеннями резервного копіювання. Система повинна бути сумісною з популярними платформами, підтримувати різні формати даних і забезпечувати високу швидкість обробки інформації. Це дозволить їй безболісно впроваджуватись у різних ІТ-середовищах.

Важливим чинником є також надійність і безпека – користувачі мають бути впевнені, що система не призведе до втрати критичних даних і забезпечує контроль цілісності копій. Для цього необхідно реалізувати ефективну систему перевірки даних і моніторингу процесів збереження.

Не менш важливою умовою успіху є якісна технічна підтримка. Наявність команди, яка швидко реагує на запити, проводить оновлення і пропонує консультації, формує позитивний імідж продукту. І, звісно, конкурентною перевагою буде економічна ефективність рішення – чим швидше система себе окупує, тим більший інтерес вона викликає серед потенційних користувачів.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

В кожній ІТ-компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [3] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яке затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотримання вимог, керівники ІТ-компаній можуть бути притягнуті до відповідальності, яка тягне накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники, то керівні особи ІТ-компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно- обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машин (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

належного освітлення. У приміщенні, в якому працюють програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

– ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.

– ризик виникнення пожежі;

– негативний вплив на органи зору людини;

– ризики ураження електричним струмом;

– недостатня, або надмірна освітленість робочого місця;

– електромагнітні (у тому числі високочастотні) випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м'язовий апарат.

Працю користувачів ЕОМ відносять до психічних форм праці з високим ступенем навантаження. Ця діяльність пов'язана зі сприйняттям зображення на екрані, постійним стеженням за його динамікою, розрізненням картин, схем, читанням тексту рукописних та друківаних матеріалів, введенням інформації з клавіатури, необхідністю підтримувати активну увагу високою ціною помилки. Будь-яка діяльність із застосуванням ЕОМ супроводжується необхідністю активації уваги та інших вищих психічних функцій, а організм людини, крім того, піддається впливу кількох десятків різноманітних факторів.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5
Довжина	6,2
Висота	3,4

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	6,2
Об'єм, V	м ³	не менше 20.0	21,8

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють двоє людей. За даними, які наведено у таблицях 8.1-8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одне робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно- обчислювальних машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-

обчислювальних машин»). Таним чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується дотримування тільки оптимальних значень показників мікроклімату.

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог. Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга). Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при нарузі вище 36 В. Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнту використання світлового потоку для приміщення ширина якого складає 5 м, довжина – 6,2 м, висота – 3,4 м.

У зазначеному приміщенні працює 5 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де

F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк; $E = 300$ Лк;

S – площа освітлюваного приміщення (у нашому випадку $S = 5 \times 6,2 = 31$ м²);

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп, обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{\text{стін}}$) і стелі ($\rho_{\text{стелі}}$), значення коефіцієнтів дорівнюють $\rho_{\text{стін}} = 50\%$ і $\rho_{\text{стелі}} = 50\%$.

Обчислимо індекс приміщення за формулою:

$$i = S / (h \cdot (A + B)),$$

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

де:

S – площа приміщення, $S = 31 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3 \text{ м}$ (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 5 \text{ м}$;

B – довжина приміщення, $B = 6,2 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекс приміщення:
 $i=0,43$.

Знаючи індекс приміщення, знаходимо $n = 0,23$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп). Підставимо всі значення у формулу, визначимо світловий потік:
 $F=66717 \text{ Лм}$.

Для розрахунку будемо використовувати світлодіодні панелі LED панель PL PFM 600 30W/3000K, світловий потік яких $F_{\text{л}} = 3000 \text{ Лм}$.

Число ламп визначається за формулою:

$$N=F/F_{\text{л}}$$

де F – світловий потік, $F_{\text{л}}$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо потрібну кількість ламп:

$$N= 66717/3000 = 22,18 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 23 шт.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи дедуплікації в інфраструктурі резервного копіювання.

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 дедуплікації в інфраструктурі резервного копіювання.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем дедуплікації в інфраструктурі резервного копіювання.

– Досліджена система дедуплікації в інфраструктурі резервного копіювання.

– На основі отриманих результатів досліджень створена програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання дедуплікації в інфраструктурі резервного копіювання.

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.

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.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

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 MARS.

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-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кривда О.С. Дослідження та програмна реалізація системи дедуплікації в інфраструктурі резервного копіювання // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

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

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

8. 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.

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

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І.,

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

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.*

16. 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.*

17. 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.*

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

19. 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.*

20. 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.*

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

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

Springer, Cham. 2021. pp 557-587.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of

					БКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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

36. 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*

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

державного технологічного університету. *Технічні науки*. №4. С. 103-110. 2020.

37. 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.

38. 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.

39. 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.

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

41. 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.

42. 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).

43. 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

44. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

45. 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.

46. 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.

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

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

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

50. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

					ВКРМ-123.25.0046.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		120