

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи віртуалізованої серверної
інфраструктури на базі HSI”

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

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
« ____ » _____ 2026 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

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

Калашніковій Карині Юріївні

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCl*

2. Керівник роботи *Смірнов Олексій Анатолійович, докт. техн. наук, професор*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 130-02 від 27.02.2026 року

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

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи віртуалізованої серверної інфраструктури на базі HCl*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Смірнов О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Калашнікова К.Ю.
(прізвище та ініціали)

АНОТАЦІЯ

Калашнікова К.Ю. Програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCl. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віртуалізованої серверної інфраструктури на базі HCl.

Метою розробки є програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCl.

Результат роботи – програмна реалізація системи віртуалізованої серверної інфраструктури на базі HCl.

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

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

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

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, віртуалізована серверна інфраструктура, HCl

ABSTRACT

Kalashnikova K.Yu. Software for a virtualized server infrastructure system based on HCI. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for a virtualized server infrastructure system based on HCI.

The purpose of the development is software for a virtualized server infrastructure system based on HCI.

The result of the work is a software implementation of a virtualized server infrastructure system based on HCI.

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

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

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

The program is developed in Python.

Keywords: computer engineering, virtualized server infrastructure, HCI

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	6
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	6
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	23
3.1 Опис функціонування системи	23
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми	28
3.4 Розробка діаграми процесів.....	33
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	36
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	36
4.2 Захист розробленого програмного забезпечення.....	54
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	57
6 ОСНОВНІ ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65

					ВКРБ-123.26.0007.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCl	Літ.	Аркуш	Аркушів
Розроб.	Калашнікова К.Ю.					Б	1	71
Перев.	Смірнов О.А.							
Н.контр.	Коваленко А.С.					ЦНТУ KI-22з		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВМ	—	віртуальні машини
ЛОМ	—	локальна обчислювальна мережа
ПК	—	персональний комп'ютер
ПЗ	—	програмне забезпечення
СЗД	—	система зберігання даних
ЦОД	—	центр обробки даних
DNS	—	Domain Name System
HCI	—	гіперконвергентна інфраструктура
http	—	HyperText Transfer Protocol
IP	—	Internet Protocol
VDI	—	технологія віртуалізації робочих місць

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Гіперконвергентна інфраструктура (HCI) трансформувала ІТ-середовища, об'єднавши обчислювальні, сховищні та мережеві ресурси в єдину систему. Дивлячись у майбутнє, HCI продовжує розвиватися, пропонуючи більш ефективні, масштабовані та гнучкі рішення для підприємств будь-якого розміру.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCI.

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

- Огляд існуючих систем віртуалізованої серверної інфраструктури на базі HCI.
- Дослідження системи віртуалізованої серверної інфраструктури на базі HCI.
- Програмна реалізація системи віртуалізованої серверної інфраструктури на базі HCI.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віртуалізованої серверної інфраструктури на базі HCI.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCI, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Технології віртуалізації робочих місць (Virtual Desktop Infrastructure, VDI) і термінального доступу дозволяють значно зменшити витрати на обслуговування користувачів і персональних комп'ютерів. Централізоване зберігання даних підвищує інформаційну безпеку компанії й скорочує ризики витоку/розкрадання важливих даних.

Всі застосунки й дані користувача зберігаються й обробляються в ЦОДі замовника, при цьому для співробітника нічого не міняється – він працює через звичний інтерфейс на своєму екрані. Подібні рішення застосовуються як у рамках окремих офісів, так і в територіально розподілених філіальних структурах з великою кількістю співробітників:

- банки;
- ритейл;
- державні організації;
- аеропорти й т.д.

Інфраструктура VDI набагато простіше й швидше в розгортанні в порівнянні із традиційними робочими станціями. Розміщення обчислювальної інфраструктури в одному ЦОД і централізоване відновлення ПЗ скорочує затримки в роботі бізнес-систем і підвищує якість роботи ІТ.

Підключення користувача до ЦОД забезпечується з використанням звичайних ПК або спеціалізованих технічних засобів – тонких клієнтів. Вони споживають набагато менше електроенергії й в окремих випадках практично не сприйнятливі до агресивних умов навколишнього середовища (підвищена температура, вологість, тривалий час роботи без перезавантаження й т.д.). Такі моделі успішно використовуються на промислових і виробничих підприємствах.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1.2 Область застосування

Гіперконвергентна інфраструктура (HCI) – це ПЗ-центрична архітектура із твердою зв'язаністю елементів зберігання, мереж і обчислень, інтегрованих у стандартне устаткування одного постачальника.

Організації розгортають гіперконвергентну інфраструктуру для своїх робітничих середовищ рівня 2 і задоволені моделлю устаткування HCI, оскільки таке рішення легко запустити. Можна почати з малого розміру й при необхідності виконати вертикальне масштабування. Можна зменшити обсяг обчислень і СЗД, щоб консолідувати робітничі середовища й знизити операційні витрати на периферії за рахунок зняття з ІТ-фахівців завдань по керуванню.

Проте HCI пов'язана з певним компромісом відносно продуктивності через втрату дискретних обчислювальних компонентів і доменів на випадок збоїв СЗД. Крім того, керованість із масштабуванням являє собою окрему проблему при вертикальному масштабуванні робітничих середовищ рівня 2 і додаванні устаткування. Можна привести такий аргумент: інфраструктура, необхідна для роботи тисяч ВМ і керування ними, повинна ставитися до робітничого середовища рівня 1, а управляти декількома великими об'єктами значно легше, ніж тисячею малих об'єктів.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи віртуалізованої серверної інфраструктури на базі HCI, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Найкращі рішення для гіперконвергентної інфраструктури

Cisco

Апаратне забезпечення, програмне забезпечення та сервісні пропозиції Cisco розроблені для забезпечення легкого доступу до інформації з будь-якого місця та в будь-який час. Інструменти гіперконвергентної інфраструктури постачальника базуються на його лінійці продуктів HyperFlex. Платформа доступна в різних конфігураціях, таких як HyperFlex Hybrid Nodes, яка призначена для розміщення робочих навантажень загального призначення. Крім того, Cisco пропонує HX220c M5, HX240c M5 та версію HX240c у великому форм-факторі. Cisco HyperFlex також удосконалена Intersight, хмарним SaaS-рішенням, яке забезпечує прогнозний аналіз збоїв, проактивне вирішення проблем, а також віддалене розгортання та управління HyperFlex Edge на периферії.

DataCore Software

Компанія DataCore Software спеціалізується на віртуалізації сховищ, управлінні сховищами та мережах сховищ. Цей постачальник дозволяє користувачам віртуалізувати сховища, розширювати ємність, покращувати доступ до даних, а також централізувати й автоматизувати адміністрування сховищ для систем Windows, UNIX, Linux, macOS та Netware. Його HCI-платформа DataCore SANsymphony в основному використовується для критично важливих, основних ІТ-ресурсів та периферійних обчислень. Нещодавно постачальник надав інтеграцію з програмним забезпеченням для резервного

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

копіювання та вдосконалення свого рішення для прогнозової аналітики DataCore Insight Services. DataCore також інтегрує гіперконвергентне сховище з мережами зберігання даних (SAN) та серверами, а також підтримує незалежне масштабування обчислювальних та сховищних ресурсів.

Dell EMC

Dell EMC забезпечує цифрову трансформацію за допомогою гібридних хмарних рішень та рішень для великих даних, побудованих на інфраструктурі центру обробки даних, яка об'єднує конвергентну інфраструктуру, сервери, сховища та технології кібербезпеки. Пристрої VxRail постачальника базуються на технології віртуальної мережі зберігання даних (vSAN) від VMware. Крім того, Dell EMC надає VMware Cloud Foundation, розроблену як готову гібридну хмарну платформу для компаній, які хочуть запускати Kubernetes у хмарному масштабі. Постачальник також надає програмне забезпечення HCI для організацій, які стандартизували технології Microsoft, а також систему Microsoft Azure Stack HCI.

Hewlett Packard Enterprise (HPE)

Hewlett Packard Enterprise (HPE) – це компанія промислових технологій, портфолію якої включає хмарні технології, центри обробки даних та робочі додатки, серед іншого. Постачальник відрізняє себе від конкурентів у сфері HCI завдяки своїй платформі дезагрегованої HCI Nimble (dHCI). Це рішення дозволяє IT-фахівцям масштабувати ресурси сховища та обчислювальні ресурси незалежно один від одного, допомагаючи своїм компаніям уникати марнотратних витрат. Nimble Storage dHCI добре підходить для організацій, яким потрібно підтримувати додатки з непередбачуваним зростанням даних, а також додатки, яким потрібен високий рівень стійкості даних або продуктивності сховища.

Infinidat

Infinidat надає підприємствам та постачальникам послуг платформенно-орієнтовану архітектуру первинного та вторинного сховища, яка забезпечує комплексні послуги обробки даних на основі InfiniVerse. Ця унікальна платформа забезпечує видатні переваги для IT-операцій, підтримку сучасних робочих

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

навантажень у локальних та гібридних багатохмарних середовищах. Кіберстійка інфраструктура Infinidat, розроблена на основі споживання, продуктивність, 100% доступність та гарантовані SLA кібербезпеки відповідають пріоритетам корпоративних ІТ та бізнесу. Відзначені нагородами платформенно-орієнтовані послуги обробки даних та визнаний сервіс «білих рукавичок» від Infinidat постійно рекомендуються клієнтами, що визнається оглядами Gartner Peer Insights.

Microsoft

Microsoft – це багатонаціональна компанія, яка розробляє, виробляє, ліцензує, підтримує та продає різноманітні програмні продукти та послуги. Постачальник пропонує корпоративні інструменти через Microsoft Azure, включаючи віртуальні машини, хмарне сховище, службу додатків та резервне копіювання в хмару. Платформа Microsoft Azure Stack HCI інтегрує локальні та периферійні служби в Azure Cloud. Azure Stack HCI пропонується як платформа на основі підписки, керована хмарою, і базується на версії Hyper-V та стеку HCI, що використовується в Azure Cloud. Платформа також пропонує спільне керування з платформою Azure Cloud та можливість використання хмарних служб Azure.

NetApp

NetApp – постачальник рішень для зберігання даних, хмарних обчислень, інформаційних технологій та управління даними. Окрім переважного пропонування локальної інфраструктури зберігання даних, постачальник також спеціалізується на гібридних хмарних сервісах обробки даних, які спрощують керування програмами та даними в хмарних та локальних середовищах з метою прискорення цифрової трансформації. NetApp використовує гіперконвергентну інфраструктуру як основу для своєї гібридної хмари. Хоча існує кілька апаратних платформ, які можна налаштувати як локальний компонент гібридного розгортання, NetApp HCI дозволяє забезпечити послідовне управління кількома хмарами.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Nutanix

Nutanix надає хмарне програмне забезпечення, обчислювальну та сховищну інфраструктуру, а також гіперконвергентні інфраструктурні рішення для впровадження віртуалізації підприємства без складного та дорогого мережевого сховища, будь то SAN чи NAS. Програмне забезпечення та послуги Nutanix HCI підтримуються в кластерах Nutanix, забезпечуючи гібридну хмарну інфраструктуру, управління та операції з єдиної консолі. Платформа також підтримує кілька гіпервізорів та охоплює низку варіантів використання завдяки інтегрованим нативним службам зберігання даних, управлінню базами даних, управлінню життєвим циклом додатків, мікросегментації, резервному копіюванню та службі аварійного відновлення. Крім того, Nutanix нещодавно залучила 750 мільйонів доларів від Bain Capital Private Equity.

Pivot3

Pivot3 пропонує динамічні гіперконвергентні та PCIe флеш-масиви, а також гібридні хмарні рішення та рішення для Інтернету речей (IoT) для забезпечення максимального використання ресурсів. Пропозиції постачальника оптимізовані для віртуалізації серверів, віртуальних робочих столів, консолідації центрів обробки даних, відеоспостереження, забезпечення безперервності бізнесу, аварійного відновлення, а також віддалених офісів та філій. HCI-платформа Pivot3, Asurity, розроблена для критично важливих, великомасштабних робочих навантажень на основі відео. Intelligence Engine постачальника також пропонує автоматизацію, штучний інтелект, автоматичне відновлення, швидке відновлення вузлів, інтелектуальний моніторинг та аналітичні функції для задоволення потреб захисту даних та доступності системи для масштабних розгортань HCI.

Sangfor Technologies

Компанія Sangfor Technologies надає рішення для IT-інфраструктури, спеціалізуючись на хмарних обчисленнях, мережевій безпеці та оптимізації мереж. Її програмна платформа HCI розроблена для модернізації центрів обробки

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

даних, корпоративних додатків, хмарної трансформації, інфраструктури віртуальних робочих столів (VDI), резервного копіювання та аварійного відновлення. Sangfor також має власний гіпервізор, що дозволяє Sangfor HCI надавати безпечну керовану платформу для малих та середніх підприємств. Постачальник також має вертикальний ринковий підхід, створюючи партнерські відносини з незалежними постачальниками програмного забезпечення та розвиваючи знання про конкретні нормативні акти, щоб задовольнити потреби урядових, медичних, виробничих та освітніх організацій.

Scale Computing

Scale Computing забезпечує гіперконвергенцію для середнього бізнесу. Платформа HyperCore3 (HC3) від цього постачальника інтегрує сервери, віртуалізацію та сховище даних в єдине рішення з високою масштабованістю. HC3 зазвичай використовується як платформа для периферійної інфраструктури або віртуалізації. Продукт Scale Computing вирізняється своїми функціями самовідновлення. HC3 автоматично виявляє проблеми та вживає коригувальних заходів без необхідності втручання людини. Крім того, рішення було розроблено як комплексна система, яку називають «центром обробки даних у коробці». Scale Computing також нещодавно розширила свою підтримку виробників серверного обладнання, випустила HE150 як недороге периферійне рішення та випустила інтегровані рішення VDI та рішення для забезпечення стійкості.

StarWind

StarWind постачає платформи гіперконвергентності та віртуалізації сховищ даних. Хоча постачальник пропонує гіперконвергентні пристрої, він також пропонує варіанти свого рішення як програмні, так і пристрої для зберігання даних, що дозволяє йому адаптуватися до будь-якого центру обробки даних. StarWind HyperConverged Appliance (HCA) в основному орієнтований на периферійні та критично важливі випадки використання для розподілених центрів обробки даних. Нещодавно постачальник додав підтримку вузлів на флеш-пам'яті, енергонезалежної пам'яті Express over Fabrics (NVMe-oF), over TCP

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

та функціональність штучного інтелекту, що надається StarWind ProActive Premium Support.

StorMagic

StorMagic пропонує рішення для зберігання даних та безпеки для підприємств з тисячами сайтів. Платформа SvSAN постачальника в основному орієнтована на периферійні та критично важливі випадки використання для розподілених центрів обробки даних. SvSAN інтегрується з існуючими віртуалізованими середовищами, дозволяє централізоване розгортання та управління, а також забезпечує роботу невеликих гіперконвергентних кластерів. StorMagic також є частиною програми HPE Complete як заміна програмного забезпечення HPE StoreVirtual. Крім того, нещодавно постачальник додав багатосайтове розгортання програмного забезпечення одним натисканням кнопки, модель ціноутворення за передплатою, хмарну технологію Witness as a Service та керування ключами як послугою.

Stratoscale

Stratoscale – це компанія, що займається хмарною інфраструктурою, яка надає комплексні програмні рішення для хмарної інфраструктури для корпоративних IT-відділів, команд розробників та постачальників послуг. Платформа Stratoscale HCI орієнтована на компанії, які хочуть створювати публічні, приватні або гібридні хмари. Рішення базується на гіпервізорі KVM та підтримує кілька архітектур сховища, включаючи блокові, файлові та об'єктні. Stratoscale також підтримує використання зовнішніх масивів сховищ, на відміну від деяких платформ HCI. Багато інструментів управління Stratoscale орієнтовані на хмару, а можливості оркестрації дозволяють адміністраторам масштабувати програми, запускаючи або видаляючи екземпляри у відповідь на зміну робочого навантаження.

VMware

VMware пропонує хмарні та віртуалізаційні послуги, що забезпечують роботу складної цифрової інфраструктури. VMware vSAN, в першу чергу, надає

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

програмну платформу HCI як компонент широкого портфолію, що охоплює управління віртуальними додатками локально, на периферії та в хмарі. Рішення HCI від VMware складається з гіпервізора VMware vSphere, vSAN, NSX та vRealize Suite. Постачальник також розширив свої можливості HCI як послуги (HCI as a Service) від AWS та IBM Cloud, щоб надати користувачам можливість запускати VMware HCI у шести найбільших гіпермасштабованих середовищах.

Приклади рішень

Як ілюстрація розглянемо кілька рішень HCI, пропонує як платформа для розгортання інфраструктури VDI.

Компанія Dell EMC розробила два комплексних рішення VDI Complete для розгортання інфраструктури віртуальних десктопів на базі технологій віртуалізації VMware. Гіперконвергентні комплекси VxRail використовують для програмно-визначаємого зберігання пакет програм Virtual SAN Enterprise, інструменти захисту даних у віртуалізованому середовищі Recovery Point for VM і vSphere Replication / Data Protection, а також утиліту для швидкого розгортання приставки VxRail Manager.

VxRail – це повністю готове до роботи рішення (за заявою вендора, розгортання всієї інфраструктури VDI за допомогою VxRail виконується менше ніж за годину). Хоча сама лінійка VxRail складається з п'яти типів вузлів, Dell EMC рекомендує для підтримки VDI вузли V Series на базі серверів 2U з повністю твердотільною (all flash) або гібридною дисковою підсистемою ємністю до 46 Тбайт на один вузол, хоча для обес-печення високої продуктивності десктопів переважно використовувати вузли в конфігурації all flash.

Друге рішення являє собою пакет для програмно-визначаємого зберігання Virtual SAN Ready Node, що поставляється разом із двосокетними серверами Dell PowerEdge R630/R730. Замовник сам повинен виконати інсталяцію на PowerEdge гіпервізора vSphere і побудувати за допомогою Virtual SAN Ready Node кластер серверів і програмно-визначаємо сховище, що складається з дисків серверів. У порівнянні з VxRail такий варіант впровадження інфраструктури VDI вимагає

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

досить високої кваліфікації ІТ-персоналу, але в той же час на устаткування буде витрачено менше засобів, а крім того, покупець зможе самостійно настроїти параметри вузлів відповідно до вимог своїх застосунків.

У якості ПЗ віртуальних робочих столів обоє рішення підтримують пакети VMware Horizon 7 і Horizon Apps, а для розгортання на віртуальних десктопах застосунків з важкою графікою можна встановити у вузлах графічні прискорювачі NVIDIA GPU (надається як опція).

Варто згадати, що Dell EMC, крім рішень для побудови апаратної платформи VDI, пропонує великий пакет тонких терміналів із серії Dell Wyse, тому замовники компанії можуть одержати повний стек рішень для впровадження інфраструктури віртуальних клієнтів.

Серед гіперконвергентних комплексів інших виробників, що просувають дане рішення у вигляді апаратної платформи для впровадження VDI, можна згадати продукти компанії Nutanix, що спеціалізується на технологіях VDI (на основі її систем випускають свої комплекси Dell EMC (серія XC) і Lenovo), а також комплекси HyperFlex від Cisco і SimpliVity від HPE. Всі ці гіперконвергентні рішення підтримують використання ПЗ віртуальних десктопів Citrix XenDesktop і VMware Horizon.

Компанія Huawei у якості готової апаратної платформи для розгортання VDI пропонує гіперконвергентний комплекс FusionCloud Appliance на базі стієчних серверів або блейд-серверів FusionCube 2000/6000/9000, що масштабується до 5000 віртуальних десктопів. Для більших інсталяцій VDI (до 20 тис. віртуальних десктопів) китайський вендор рекомендує використовувати інтегровану систему FusionCloud Standard, що складається із блейд-серверів E9000 і окремого дискового масиву середнього класу S5500T.

Інший шлях до скорочення первісних витрат на побудову інфраструктури віртуальних десктопів вибрала компанія Parallels, що розробила рішення Parallels Containers for Windows (PCW) для контейнеризації ОС Windows у середовищі Windows Server. У порівнянні із традиційним підходом до реалізації VDI, у

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

випадку застосування контейнерів Windows щільність розгортання на сервері віртуальних десктопів істотно збільшується (до 200-250 штук на один сервер), що знижує вимоги до потужності серверної платформи VDI, а також дозволяє заощадити на ліцензуванні Microsoft Windows.

Український системний інтегратор IBS разом з Parallels розробив на базі PCW три додаткових компоненти, що спрощують розгортання інфраструктури віртуальних робочих столів: брокер з'єднань для керування інфраструктурою VDI і підключення до неї користувачів Odin Connection Broker, клієнт для пристроїв доступу під керуванням Linux і Windows, призначений для підключення до віртуальних робочих місць Odin VDI Client, а також програму-агент Odin Hardware Node Extention, що дозволяє брокерові з'єднань управляти хостами віртуалізації.

В основі платформи Odin VDI лежить стандартна серверна операційна система Windows Server, поверх якої встановлюється PCW і запускаються робітничі середовища, так звані контейнери, ізольовані на рівні адресного простору оперативної пам'яті й закритої (private) області на диску. Контейнери не мають потреби в індивідуальній інсталяції операційної системи й емуляції роботи устаткування, оскільки в усіх запускається екземпляр базової ОС. У кожного з них свої користувачі, у кожному виконуються індивідуальні процеси, є власний реєстр, мережні адаптери, виділений обсяг оперативної пам'яті, а також жорсткий диск. За заявою розроблювачів, у режимі очікування контейнер займає всього 200-300 Мбайт оперативної пам'яті.

Гіперконвергентні рішення Dell EMC

Вигода від розширення портфеля HCI компанії Dell EMC. Перешикуйте вашу роботу за допомогою інтегрованих систем під ключ, які прискорюють модернізацію IT, або продовжуйте використовувати існуючі операційні моделі, одержуючи при цьому переваги HCI за допомогою гнучких, попередньо перевірених будівельних блоків HCI.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Dell EMC VxRail

Створені для VMware, разом з VMware і для поліпшення VMware, системи VxRail надають найшвидший і простий шлях до трансформації й модернізації інфраструктури центра обробки даних. VxRail – єдина платформа гіперконвергентної інфраструктури для кожного робочого навантаження й сценарію використання VMware, включаючи VDI, застосунка з високою обчислювальною щільністю, а також для традиційних і хмарних застосунків у справжній гібридній хмарній інфраструктурі.

VxRail HCI System Software – перша і єдина ОС для гіперконвергентної інфраструктури, розроблена разом з VMware. Вона забезпечує ефективність експлуатації, цілісний стік і автоматизоване комплексне керування життєвим циклом, а також прогнозу аналітикові на основі ШІ.

Dell EMC доставляє систему VxRail попередньо налаштовану й протестовану в обраній вами конфігурації. Починаючи від пристроїв до інтегрованих стійок, з мережним устаткуванням або без, ви зможете підібрати рішення VxRail, що оптимально відповідає вашим потребам.

Основні характеристики:

- Програмно-визначаєма архітектура, що консолідує ресурси обчислень, зберігання, віртуалізації й керування.
- Інтеграція з усім стеком технологій VMware, включаючи VMware Cloud Foundation.
- Надає сервіси керування даними, забезпечує відказостійкість і якість обслуговування корпоративного рівня.
- Автоматизація налаштування мережі за допомогою функції SmartFabric Services for VxRail, що дозволяє значно спростити й прискорити розгортання.
- Єдина крапка підтримки для програмного й апаратного забезпечення.
- Analytics Consulting Engine (ACE) забезпечує практично значиму аналітику й керування декількома кластерами з машинним навчанням.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Dell EMC Cloud for Microsoft Azure Stack

Dell EMC Cloud for Microsoft Azure Stack – це локальна гібридна хмарна платформа для надання інфраструктури й моделі «платформа як послуга» (PaaS) з погодженими умовами роботи в хмарі Azure на локальній площадці або у віддалених середовищах.

Виконайте цифрову трансформацію бізнесу, використовуючи погоджені з Azure сервіси, хмарні застосунки й ресурси для самообслуговування. Забезпечте швидке, просте й упевнене впровадження гібридної хмари.

Основні характеристики:

- Локальна й периферійна платформа гібридної хмари.
- Автоматизоване надання ІТ-послуг.
- Спеціалізовані хмарні застосунки.
- Прискорене впровадження гібридної хмари.
- Ресурси для самообслуговування на вимогу.
- Погоджені умови роботи на платформі Azure на локальній площадці або в периферійних середовищах.

Ефективно надавайте застосунки й послуги завдяки платформі з погодженим інтерфейсом Azure для локального середовища:

- Забезпечте більше швидке надання застосунків завдяки ресурсам, доступним на вимогу в режимі самообслуговування.
- Впровадьте інновації за допомогою хмарних застосунків, щоб виконати цифрову трансформацію вашого бізнесу.
- Забезпечте швидке, просте й упевнене впровадження гібридної хмари.
- Гнучкі варіанти фінансування, доступні в рамках моделі Flex on Demand.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

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

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

Переваги та недоліки Python

Переваги:

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

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

потрібно витрачати багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

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

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

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

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

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

Недоліки:

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

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

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

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

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

Застосування Python:

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

– Для візуалізації даних. Візуалізація даних – це окрема частина аналізу даних, яка допомагає нам подавати інформацію, необроблену чи очищену, у більш змістовній формі. Тут Python знову входить у гру, пропонуючи широкий спектр інструментів візуалізації даних. Найпопулярніші з них – matplotlib і

заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

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

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок, тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

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

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати,

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи віртуалізованої серверної інфраструктури на базі HCl.

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

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Рішення гіперконвергентної інфраструктури (HCI) забезпечують обчислення, зберігання даних, мережеве керування та керування хмарою з єдиного екземпляра, що працює на серверному обладнанні. Програмне забезпечення HCI спочатку використовувалося як альтернатива дорогим та складним масивам зберігання даних для середовищ VMware. Ці інструменти, які раніше називалися гіперконвергентними інтегрованими системами (HCIS), переважно продавалися як пристрої одним постачальником обладнання. Однак, з розвитком ринку, виникла різниця між HCIS та HCI.

Усі рішення HCI включають певний набір можливостей, включаючи віртуальні обчислення, сховище та мережу, використовуючи масштабовану архітектуру без спільного використання. Крім того, програмне забезпечення HCI включає уніфіковане керування, локальне сховище з прямим підключенням (DAS) у кожному вузлі, високу доступність, служби обробки даних та певний рівень вибору серверного та мережевого обладнання. Деякі з цих функцій можуть бути реалізовані гіпервізорами, інструментами управління та мережевими можливостями або через платформу зберігання даних у поєднанні зі сторонніми постачальниками програмного забезпечення для завершення програмного стеку. Крім того, деякі постачальники HCI почали пропонувати хмарні послуги.

За суттю, гіперконвергентна інфраструктура інтегрує обчислювальні, сховищні та мережеві ресурси в єдиний програмно-керований пристрій. Це відрізняється від традиційної конвергентної інфраструктури, де ці компоненти є окремими, але об'єднаними. Ключ до успіху HCI полягає в її програмному рівні, який абстрагує та об'єднує ресурси між вузлами, що забезпечує спрощене управління та масштабованість.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Ключові компоненти архітектури HCI:

- Обчислювальні вузли: це сервери, що забезпечують обчислювальну потужність. Сучасні HCI-рішення часто використовують сервери на базі x86 від таких постачальників, як Dell, HPE або Cisco.
- Гіперконвергентне сховище: Сховище віртуалізовано та розподілено по кластеру, що усуває потребу в окремих пристроях SAN або NAS. Цей гіперконвергентний рівень сховища підвищує продуктивність та стійкість.
- Мережі: Хоча мережеве обладнання залишається окремим, програмне забезпечення HCI ефективно керує мережевим трафіком, часто інтегруючись із рішеннями програмно-визначених мереж (SDN).
- Програмне забезпечення для керування: такі платформи, як Nutanix AOS або VMware vSAN, забезпечують централізоване керування, автоматизацію та моніторинг.

Еволюція до хмарно-готової HCI

До 2026 року гіперконвергентна інфраструктура перейме на основі хмарних принципів. Багато платформ HCI тепер підтримують оркестрацію контейнерів за допомогою Kubernetes, що дозволяє розгортати гібридні хмари. Цей перехід дозволяє організаціям безперешкодно запускати традиційні віртуальні машини разом із контейнерними робочими навантаженнями.

Варіанти використання, що сприяють впровадженню HCI у 2026 році

Гнучкість та простота гіперконвергентної інфраструктури роблять її придатною для широкого спектру сценаріїв. Ось деякі з найпоширеніших випадків використання, які я бачив у нещодавніх розгортаннях:

1. Модернізація центру обробки даних. Багато підприємств замінюють застарілу конвергентну інфраструктуру або ізольовані системи зберігання та обчислення на HCI-рішення. Наприклад, гіперконвергентні інфраструктурні пристрої Dell VxRail пропонують тісну інтеграцію із середовищами VMware, що робить їх популярним вибором для оновлення центрів обробки даних.

2. Периферійні обчислення. Зростання популярності Інтернету речей (IoT)

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

та віддаленої роботи збільшило попит на периферійні обчислення. HCI-сервери від таких постачальників, як Nutanix та HPE Simplivity, забезпечують компактні, стійкі платформи, які можна розгортати у філіях або роздрібних магазинах, забезпечуючи локальну обчислювальну потужність із централізованим керуванням.

3. Інфраструктура віртуальних робочих столів (VDI). Робочі навантаження VDI отримують вигоду від масштабованості та продуктивності гіперконвергентного сховища в поєднанні з обчислювальними ресурсами. Наприклад, рішення VMware для гіперконвергентної інфраструктури широко використовуються для підтримки тисяч віртуальних робочих столів із узгодженим користувацьким інтерфейсом.

4. Відновлення після збоїв та резервне копіювання. Гіперконвергентна інфраструктура спрощує аварійне відновлення, забезпечуючи реплікацію між сайтами без складних конфігурацій сховища. Гіперконвергентні рішення Dell EMC часто включають вбудовані функції захисту даних, які скорочують цільовий час відновлення (RTO).

3.2 Розробка структурної схеми

Гіперконвергентна інфраструктура (HCI) поєднує обчислювальні, мережні ресурси й ресурси зберігання в єдину систему. Це оптимізоване рішення на базі ПЗ й серверів x86 заміняє спеціалізоване устаткування. Гіперконвергентна інфраструктура дає можливість спростити середовище ЦОД і поліпшити його масштабованість. З мого досвіду роботи з різними організаціями, виявилось кілька шаблонів розгортання, які підкреслюють ефективність використання HCI.

Масштабовані кластери

Більшість розгортань HCI починаються з невеликих вузлів, і масштабуються в міру зростання попиту. Такий модульний підхід дозволяє компаніям збільшувати потужності без простоїв або складної реконфігурації.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Наприклад, роздрібна мережа може почати з кластера з 3 вузлів у головному офісі та розширити його до 10 вузлів у міру додавання нових магазинів.

Гібридна хмарна інтеграція

Багато компаній інтегрують свою гіперконвергентну інфраструктуру з публічними хмарними сервісами. Ця гібридна модель підтримує мобільність робочих навантажень та аварійне відновлення. Наприклад, гіперконвергентна інфраструктура Nutanix пропонує безперешкодну інтеграцію з Azure та AWS, що дозволяє створювати гібридні хмарні середовища.

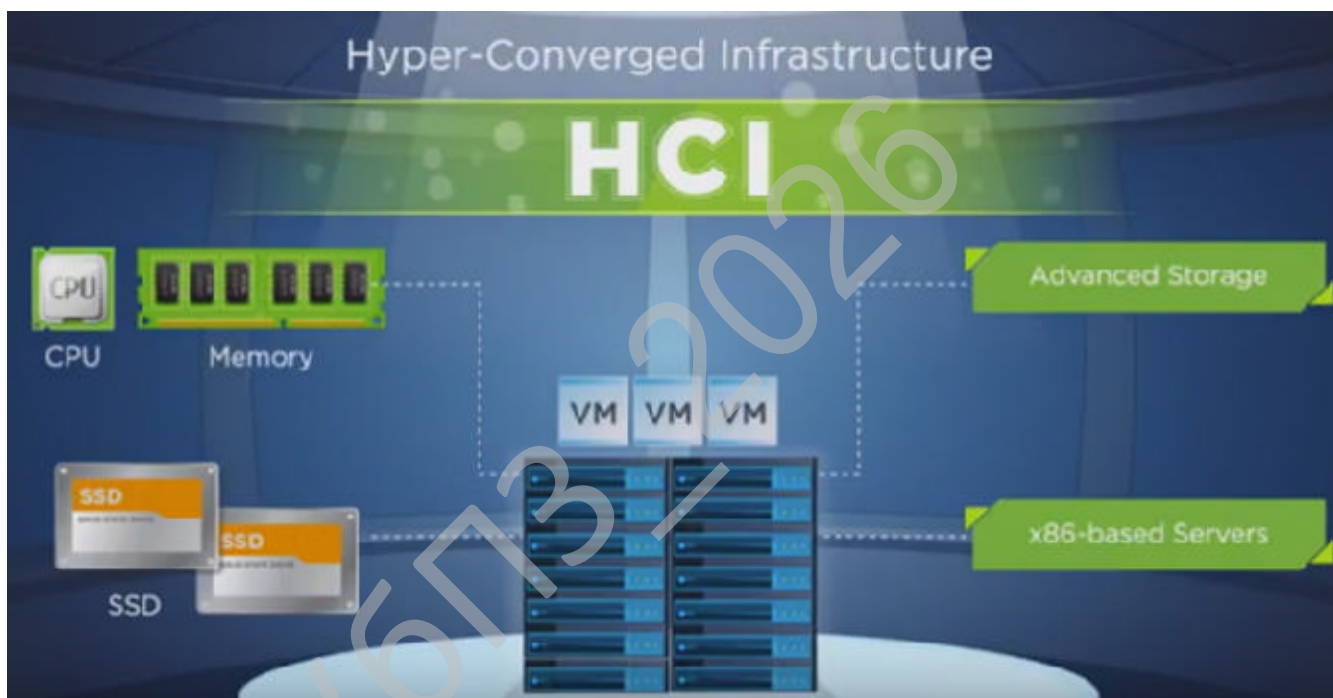


Рисунок 3.1 – Структурна схема системи

Консолідація кількох робочих навантажень

Організації часто консолідують різноманітні робочі навантаження на одній платформі HCI. Це включає бази даних, веб-сервери, аналітику та віртуальні робочі столи. Гіперконвергентна архітектура підтримує це, забезпечуючи стабільну продуктивність та спрощене управління в різних програмах.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Екосистема постачальників та підтримка

Вибір правильного постачальника гіперконвергентної інфраструктури є критично важливим. Dell, Nutanix, HPE та Cisco є одними з провідних гравців, кожен з яких пропонує унікальні функції та інтеграції. «Магічний квадрант» Gartner для програмного забезпечення для гіперконвергентної інфраструктури залишається цінним ресурсом для оцінки цих постачальників.

Проблеми та міркування

Хоча гіперконвергентна інфраструктура пропонує багато переваг, важливо враховувати деякі проблеми:

- Вартість: Початкові інвестиції можуть бути високими, особливо для малого бізнесу. Однак загальна вартість володіння часто зменшується з часом завдяки спрощенню управління.
- Прив'язка до постачальника: деякі платформи HSI значною мірою залежать від власного програмного забезпечення, що може обмежувати гнучкість.
- Вимоги до навичок: IT-командам потрібне навчання для ефективного управління гіперконвергентними середовищами, особливо під час інтеграції з хмарними сервісами.

Погляд у майбутнє: майбутнє HSI

Майбутнє гіперконвергентної інфраструктури вказує на глибшу інтеграцію з хмарними технологіями, управління на основі штучного інтелекту та підвищену автоматизацію. Проекти гіперконвергентної інфраструктури з відкритим кодом набирають обертів, пропонуючи організаціям більше можливостей для налаштування. У 2026 році я очікую побачити більше гібридних та багатохмарних розгортань HSI, при цьому постачальники розширюватимуть підтримку контейнерних робочих навантажень та периферійних обчислень. Основна увага буде приділена створенню хмарної, масштабованої та простої в управлінні інфраструктури, яка підтримує ініціативи цифрової трансформації.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3.3 Розробка функціональної схеми

Гіперконвергенція – об'єднання ІТ-ресурсів у цілісне рішення за допомогою типових апаратних блоків і програмно обумовлених пристроїв – стала одним із самих модних трендів в ІТ. Спробуємо розібратися в перевагах гіперконвергентних систем, напрямках подальшого розвитку рішень і сценаріях їхнього використання в бізнесі.

Етапи розвитку обчислювальних інфраструктур

Віртуалізація машин, що застосовувалася ще в радянський час, дозволила оптимізувати використання обчислювальних ресурсів комп'ютера, відокремити виконання коду від конкретного апаратного забезпечення й створила передумови до виникнення масштабованих віртуальних обчислювальних потужностей на blade-серверах, що містять, що додаються по необхідності «леза» – материнські плати із процесорами, установлювані в blade-кошики.

На початку 90-х з'явилася можливість нарощувати обсяг дискового простору шляхом додавання дисків у виділене сховище даних – СЗД. Для роботи з ними використовувалася розроблена в 1987 році технологія віртуалізації даних RAID, що поєднує диски в логічні елементи, з якими й працювали операційні системи.

Архітектура, що включає кластер комп'ютерів або блейд-серверів і пов'язаних з ними швидкісним каналом (fibre channel) СЗД, де-факто стала стандартом в 2005 році й донині використовується багатьма компаніями.

Цифрова трансформація ставить перед розроблювачами комп'ютерних систем всі нові завдання. Складність рішень постійно росте, обчислювальні системи потрібно запускати швидко, забезпечуючи необхідну гнучкість при подальших змінах конфігурації. У відповідь на запити ринку з'явилися хмарні технології, що надають обчислювальні потужності як сервіс на вимогу, і конвергентні рішення – заздалегідь інтегровані обчислювальні модулі, що

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

включають обчислювальні потужності, системи зберігання, мережне устаткування, засоби віртуалізації й оркестрації.

Раніше компанія купувала комп'ютери й СЗД, з'єднувала їх і розвертала платформу віртуалізації. Це вимагало багато часу й участі кваліфікованих фахівців. З появою конвергентних систем (у термінології Gartner – інтегрованих інфраструктур) стало досить купити заздалегідь інтегрований модуль (стійку), готовий для установки прикладних програм.

Наступний крок – розгорнути платформу віртуалізації на інтегрованих блоках або серверах, об'єднавши в єдине ціле їхні внутрішні диски/дискові полки на програмному рівні, у такий спосіб формуючи віртуальне відказостійке сховище. З погляду нарощування обчислювальних потужностей гіперконвергентні інфраструктури, що виходять (HCI – Hyper-Converged Infrastructure) уже не обмежені можливостями одного інтегрованого модуля пристрою або сервера – платформа поєднує їхні ресурси в єдине ціле, з якого можна нарізати потрібні для конкретного завдання обчислювальні потужності.

Весь «інтелект» HCI для забезпечення відказостійкості й продуктивності опирається на програмно обумовлені технології (віртуальні машини, програмно обумовлені сховища).

Такий підхід сильно спрощує керування складною системою, що тепер здійснюється через загальну консоль адміністрування, забезпечує більшу гнучкість і масштабованість, а також виконання заздалегідь заданих вимог до відказостійкості й продуктивності. При цьому можна обійтися без дорогих СЗД, відмовитися від мереж fiber channel і зовнішніх мереж зберігання даних – гіперконвергентні системи, як правило, складаються з комодіті-рішень, побудованих на стандартній для галузі архітектурі x86, що значно підвищує економічну ефективність їхнього використання.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29



Рисунок 3.2 – Функціональна схема системи

Сценарії застосування

HCI не є рішенням на всі випадки життя. Масштабування може вийти не збалансованим: наприклад, клієнтові потрібно тільки наростити обчислювальну потужність, а доводиться докуповувати весь блок. Крім того, є проблеми з розгортанням деяких застосунків у віртуальних середовищах.

Два основних сценарії використання гіперконвергентних систем – віддалені офіси й віртуалізовані робітники місця. При гомогенному навантаженні, як у випадку VDI, проявляється основна перевага

гіперконвергентної системи – легка масштабованість. Допустимо, сьогодні потрібно 1 тис. робочих місць – купуємо десять вузлів HCI. Завтра потрібно 1,5 тис. – купуємо ще п'ять. У такий спосіб сильно спрощується розгортання обчислювальної інфраструктури віддаленого офісу або філії. При цьому не треба шукати місцевих фахівців для запуску й підтримки системи, тим більше що їх може й не бути.

Втім, можливі й інші сценарії застосування. Наприклад, об'єднання різнорідних обчислювальних систем при злитті компаній легше здійснити при інтеграції на HCI-системах. Та й не тільки при об'єднанні – міграція с дорогих високонадійних обчислювальних систем, на яких працюють бізнес-критичні застосунки, на комодіті-рішення, використовувані в HCI, може заощадити багато грошей, не знижуючи при цьому показники надійності.

І, звичайно, гнучкі гіперконвергентні системи зручні для розробки й тестування нових бізнес-застосунків.

Перехід на HCI

Найпростіший спосіб перейти на гіперконвергентну систему – купити заздалегідь інтегрований програмно-апаратний комплекс. Такий підхід ефективний при розгортанні нових філій підприємств і банків, створенні нових компаній.

Але вже працюючі на ринку організації мають свою IT-інфраструктуру й не хочуть втрачати зроблені в неї інвестиції. У компаній існують програми закупівель, орієнтовані на конкретних вендорів, є експертиза по роботі з устаткуванням і програмним забезпеченням конкретних виробників.

Постачальники у відповідь на запит ринку почали пропонувати різні варіанти гіперконвергентних систем. Постачальники апаратних рішень забезпечують можливість роботи з різними системами віртуалізації. Так, Dell EMC пропонує гіперконвергентні системи на базі VMware, Nutanix і Azure Stack, але може поставити сервера й з усім інтегрованим стеком Red Hat. Постачальники програмних рішень також провели диверсифікованість апаратних

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

платформ. Написати програму, що працює з конкретним апаратним комплексом, значно простіше, ніж забезпечити її роботу з усім спектром «заліза», представленим на ринку. Це завдання різного масштабу. Рішення подорослішали й тепер з певними обмеженнями можуть бути запуснені практично на будь-якому стандартному устаткуванні. Але, як відзначив експерт по хмарних платформах компанії Dell EMC, отут є підводні камені. Робота 3D залежить від особливостей апаратної частини й версій її прошивання. Наприклад, продукт Dell EMC ScaleIO працює через гіпервізор і мало пов'язаний з «залізом». Але з певного часу його перестали продавати як окреме ПЗ, оскільки клієнти іноді встановлювали його на екзотичне самозбірне «залізо», а потім висували претензії компанії Dell EMC. Тому було ухвалене рішення продавати ScaleIO тільки на сертифікованому устаткуванні, по суті по моделі гіперконвергентної інфраструктури VxRail. Проводять сертифікацію конкретних серверів для своїх гіперконвергентних рішень компанії VMware і Red Hat.

У єдине ціле можна об'єднати не тільки обчислювальні потужності й системи зберігання даних, але й віртуалізовані мережі, що вже стає стандартом для HCI. Пропонуються рішення, інтегровані із системами резервування й аварійне відновлення даних.

Так, в Dell EMC крім класичної гіперконвергентної інфраструктури VxRail з'явилося більше складне гіперконвергентне рішення Dell EMC VxRail SDDC на базі VMware Cloud Foundation, куди входять віртуальне сховище VSAN і програмно обумовлені мережі VMware NSX. Рішення розширюється практично будь-якими продуктами VMware, включаючи систему автоматизації керування ІТ-процесами VMware vRealize Operations, систему створення персоналізованої інфраструктури, застосунків і спеціалізованих ІТ-послуг VMware vRealize Automation. Причому все це може поставлятися як попередньо інтегрований комплекс «з коробки».

Таким чином, гіперконвергентні рішення виходять на новий рівень. Якщо раніше пропонувалася тільки інфраструктура, на якій фахівці замовника

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

«піднімали» свої системи, то тепер гіперконвергентні рішення пропонуються як платформа, на зразок хмарного PaaS. Це сильно полегшує праця розроблювачів, дозволяючи автоматизувати створення середовищ виконання програм, розгортання віртуальних машин, контейнерів і мікросервісів.

Ще один напрямок розвитку гіперконвергентних систем – надання їхнім хмарним провайдером в оренду по моделі dedicated. Компанія може не купувати HCI-систему, а користуватися нею із ЦОДу провайдеру по сервісній моделі. Або розгорнути в себе й масштабувати в ЦОД провайдеру, створюючи розтягнутий кластер. Аналогічний кластер можна створити, об'єднавши зі своєї локальної HCI-системою гіперконвергентну, розташовану по моделі colocation у ЦОДі провайдеру.

На Заході вже з'явилися пропозиції as-a-Service, наприклад на базі NetApp HCI. Це дозволить клієнтам створити гіперконвергентні рішення із хмарою VMware, об'єднавши локальні й хмарні віртуалізованні системи зберігання за допомогою vSAN у розтягнутий кластер

Сьогодні обчислювальна інфраструктура підприємств містить у собі усе більше різноманітних систем і її розвиток спрямований на перехід від забезпечення працездатності обчислювальних систем до забезпечення доступності й працездатності пропонованих клієнтові сервісів. У цих умовах одне з головних переваг гіперконвергентних систем – спрощення керування обчислювальною інфраструктурою, причому як на площадці замовника, так і в провайдеру.

3.4 Розробка діаграми процесів

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

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

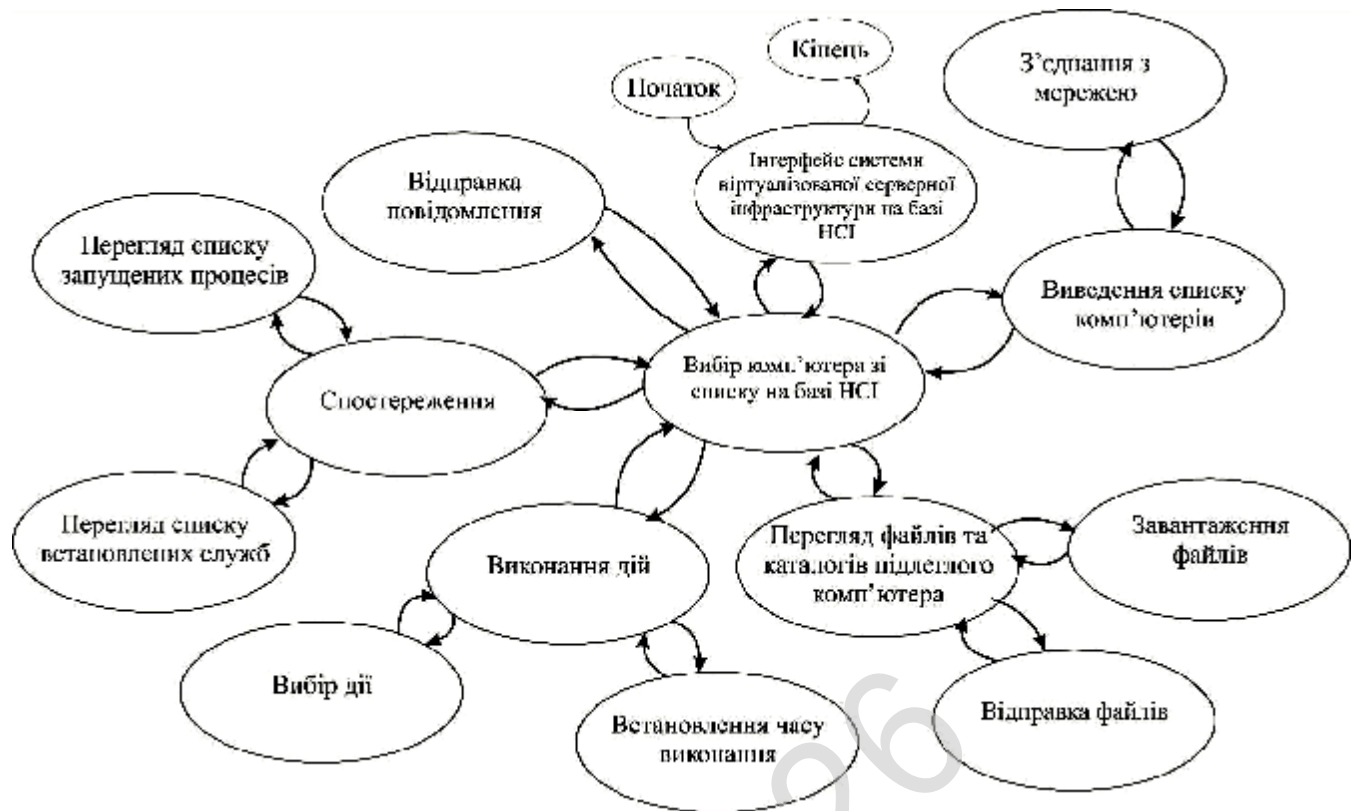


Рисунок 3.3 – Діаграма взаємодії процесів

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

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

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

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

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

Діаграми потоків даних містять чотири типи елементів:

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

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.

К6ПЗ_2026

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

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

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

```
graph TD
    Start([Початок]) --> Init[Ініціалізація змінних  
системи віртуалізованої серверної  
інфраструктури]
    Init --> M1((M1))
    M1 --> Display{Відправка файлу?}
    Display -- ТАК --> Send[Відправка вказаного  
файлу на вибраний  
комп'ютер]
    Display -- НІ --> Select{Вибір файлу з  
папки?}
    Select -- ТАК --> Print[Відправка вказаного  
файлу на друк на  
принтер вибраного  
комп'ютера]
    Select -- НІ --> M2((M2))
    M2 --> ShowList[/Виведення  
списку підключених  
комп'ютерів/]
    ShowList --> SelectComp{Вибрати комп'ютер  
зі списку?}
    SelectComp -- ТАК --> ShowFiles{Показати файли на  
виступі?}
    ShowFiles -- ТАК --> ShowDetails[/Відобразити  
список підключених  
дисків та  
розміщених на них  
файлів і каталогів/]
    ShowDetails --> SelectFile{Завантажити файл?}
    SelectFile -- ТАК --> Download[Завантаження  
вибраного файлу]
    Download --> M1
    SelectFile -- НІ --> M2
    M2 --> SelectDir{Вибір директорії  
для розміщення файлу?}
    SelectDir -- ТАК --> CreateDir[Створення директорії  
у вказаній папці]
    CreateDir --> SelectFile
    SelectDir -- НІ --> M2
    M2 --> SelectAction{Вибір дії для  
виконання на базі HCL}
    SelectAction --> Wait[/Введення часу  
виконання/]
    Wait --> ShowMsg[/Введення  
повідомлення, що буде  
випущено на  
підключений комп'ютер  
під час виконання  
вибраної дії/]
    ShowMsg --> Execute[Виконати вказану дію]
    Execute --> EndCheck{Закінчення обробки  
повідомлення?}
    EndCheck -- ТАК --> End([КІНЕЦЬ])
    EndCheck -- НІ --> M2
```

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

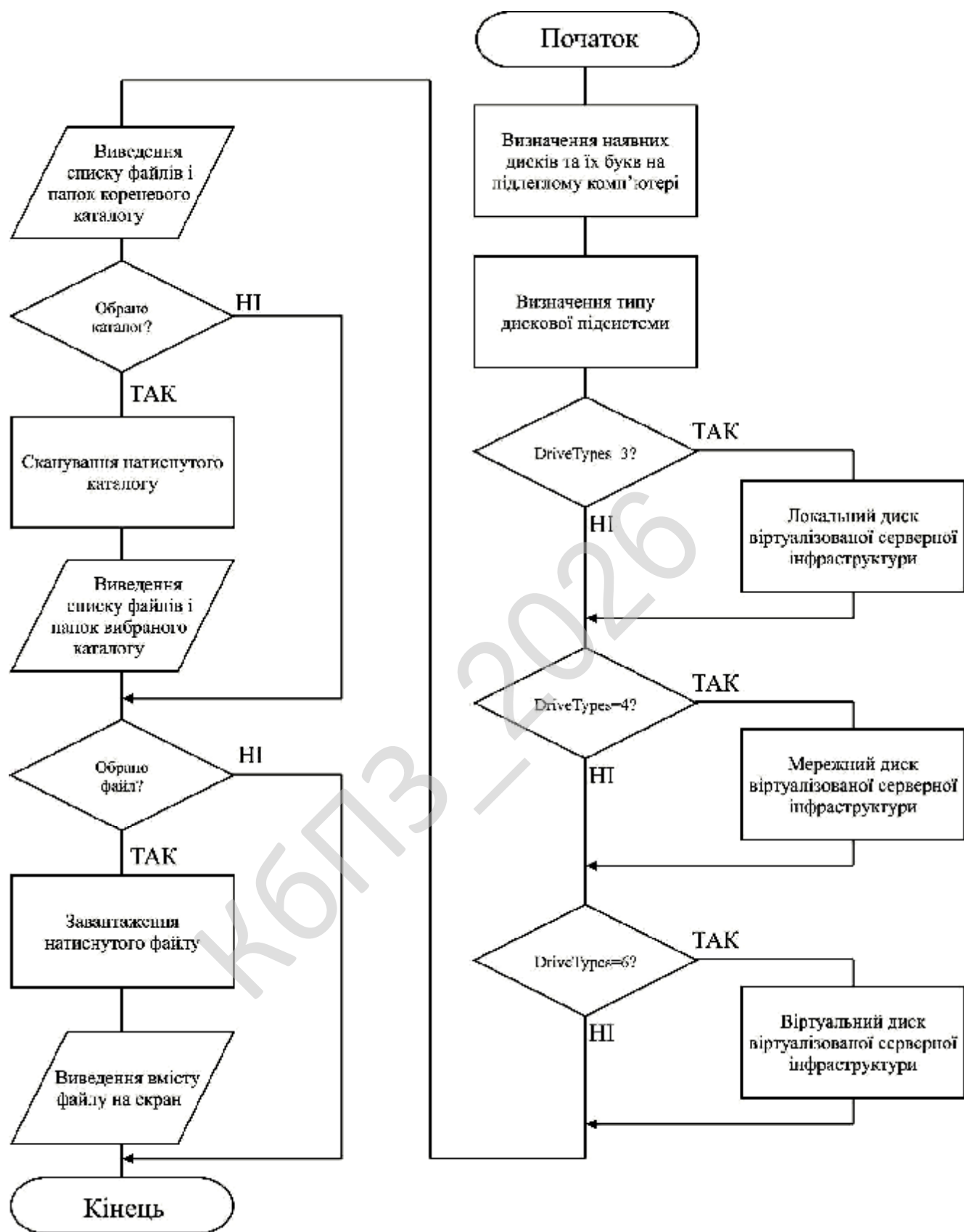


Рисунок 4.2 – Блок-схема роботи підпрограми

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

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

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

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

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

У 1991 був утворений консорціум ODMG, основною метою якого стало вироблення промислового стандарту об'єктно-орієнтованих баз даних. Між 1993

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

та 2001 роками ODMG опублікувала п'ять ревізій своїх специфікацій. Остання версія стандарту має індекс 3.0, після чого група розпустилася. До кінця 1990-х існувало близько десяти компаній, що виробляли комерційні продукти, що позиціонуються на ринку як ООСКБД. Найбільш відомими системами даного класу стали Objectivity, Versant виробництва однойменних компаній, а також СКБД Jasmine, випущена компанією CA. Незважаючи на переваги, що дозволяють ефективніше вирішувати певний ряд завдань, об'єктно-орієнтовані системи так і не змогли завоювати значущу частку ринку СКБД, залишившись «нішевим» продуктом.

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

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

Поняття СКБД третього покоління, якими, власне кажучи, і є об'єктно-реляційні СКБД, з'явилося після опублікування групою відомих фахівців в області баз даних «Маніфесту систем баз даних третього покоління». Основні принципи СКБД третього покоління, позначені в маніфесті:

Крім традиційних послуг з управління даними, СКБД третього покоління повинні забезпечити підтримку розвиненіших структур об'єктів і правил. Розвинутіша структура об'єктів характеризує засоби, необхідні для зберігання і маніпулювання нетрадиційними елементами даних (тексти, просторові дані, мультимедіа).

СКБД третього покоління повинні включити в себе СКБД другого покоління. Системи другого покоління внесли вирішальний вклад у двох областях – непроцедурний доступ за допомогою мови запитів SQL і незалежність

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

даних. Ці досягнення обов'язково повинні враховуватися в системах третього покоління.

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

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

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

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

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

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.

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

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

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

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

Основні характеристики СКБД

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

Можливості СКБД

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

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

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

Основні компоненти середовища СКБД

1. Апаратне забезпечення.

2. Програмне забезпечення.

3. Дані.

4. Процедури – інструкції та правила, які повинні враховуватись при проектуванні та використанні БД.

5. Користувачі: адміністратори даних (керування даними, проектування БД, розробка алгоритмів, процедур) та БД (фізичне проектування, відповідальність за безпеку та цілісність даних); розробники БД; прикладні програмісти; кінцеві користувачі.

Архітектура СКБД

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

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

2. Концептуальний – узагальнене представлення БД, описує які дані зберігаються в БД і зв'язки між ними. Підтримує зовнішні представлення, підтримується внутрішнім рівнем.

3. Внутрішній – фізичне представлення БД в комп'ютері.

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

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

Було використано **MongoDB** – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій.

У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу.

Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

При розробці автори виходили з необхідності спеціалізації баз даних, завдяки чому їм вдалося відійти від принципу «один розмір під усе». За рахунок мінімізації семантики для роботи з транзакціями з'являється можливість вирішення цілого ряду проблем, пов'язаних з нестачею продуктивності, причому горизонтальне масштабування стає простішим. Використовувана модель документів зберігання даних (JSON/BSON) простіше кодується, простіше управляється (у тому числі за рахунок застосування так званого безсхемного стилю (schemaless style), а внутрішнє угруповання релевантних даних забезпечує додатковий вииграш в швидкодії.

Нереляційний підхід досить зручний для створення баз даних, у яких горизонтальне масштабування означає розгортання на множині машин. Можливість забезпечувати найкращу продуктивність повинна існувати паралельно з підтримкою більшої функціональності, ніж це дозволяє використання пар «ключ-значення» (у чистому вигляді).

Технологія баз даних має працювати скрізь, починаючи з серверів користувача та віртуальних машин і закінчуючи хмарними технологіями.

MongoDB, на думку розробників, має заповнити розрив між простими сховищами даних типу «ключ-значення» (швидкими і легко масштабованими) і великими СКБД (зі структурними схемами і потужними запитами).

Основні можливості MongoDB:

1. Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних).
2. Досить гнучка мова для формування запитів.
3. Динамічні запити.
4. Повна підтримка індексів.
5. Профілювання запитів.
7. Швидкі оновлення «на місці».

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

8. Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео.

9. Журналювання операцій, що модифікують дані в БД.

10. Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг.

11. Може працювати відповідно до парадигми MapReduce.

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS.

Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Є докладна і якісна документація, велике число прикладів і драйверів для популярних мов Java, C++, C#, PHP, Python, Perl, Ruby.

При випуску одразу було заявлено, що реліз MongoDB 1.0 готовий до використання у виробництві як одиничний хост, так і у зв'язках master/slave.

Код цього релізу досить стабільний і перевірений в промисловій експлуатації протягом 1,5 років. MongoDB рекомендується розгортати мінімум на двох серверах використовуючи реплікацію Master/Slave.

Це забезпечує наявність актуальних даних при виході з ладу однієї з СКБД. MongoDB – продукт досить молодий, і відтак у ньому зустрічаються помилки, з'являються нові можливості тощо. Характерний високий темп розробки (проект пишуть не тільки волонтери, а й компанія людей на повній зайнятості). Компанія-розробник надає платні підтримку, хостинг, консультації.

Приклади запитів

Запити можуть витягати дані з вбудованих об'єктів та масивів. Якщо в колекцію users вставлений такий об'єкт:

```
{
  "username" : "bob",
  "address" : {
    "street" : "123 Main Street",
    "city" : "Springfield",
```

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

```

    "state" : "NY"
  }
}

```

Ми можемо запитати цей документ (і всі документи з адресою в Нью-Йорку) за допомогою:

```
> db.users.find({"address.state" : "NY"})
```

Можна також запитати елементи масиву:

```
> db.food.insert({"fruit" : ["peach", "plum", "pear"]})
> db.food.find({"fruit" : "pear"})
```

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко, використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу.

Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППІ, як правило, пов'язаний із бібліотеками програмного забезпечення: ППІ описує і вказує очікувану поведінку в той час, як бібліотека є фактичною реалізацією даного набору правил. Один ППІ може мати декілька реалізацій (або жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППІ, але на відміну від звичайного використання ППІ, доступ до поведінки вбудований в платформу

опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента.

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

Коли ППІ реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППІ може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППІ, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППІ, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки.

Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППІ визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

(REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA). Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

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

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

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

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

Як розширення концепції швидкої розробки додатків, DSDM фокусується на проектах інформаційних систем, що характеризуються стислими термінами і бюджетами. У DSDM присутні вказівки на типові помилки проектів інформаційних систем, таких як перевищення бюджету, запізнення з термінами

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

здачі (виконання), недостатнє залучення користувачів і топ-менеджерів в роботу над проектом. DSDM складається з:

- трьох стадій: передпроектна стадія життєвого циклу проекту і постпроектна стадія;

- стадія життя проекту складається з 5 етапів: дослідження реалізованості, дослідження економічної доцільності, створення функціональної моделі, проектування і розробка, етап реалізації.

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

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

На початку 1990-х в індустрії інформаційних технологій став поширюватись новий термін – швидка розробка додатків (Rapid Application Development, RAD). Інтерфейси прикладних програм еволюціонували від старих зелених екранів до графічних інтерфейсів користувача, які використовуються і зараз. На ринок почали виходити нові інструменти для створення додатків, наприклад PowerBuilder. Вони дозволили розробникам простіше ділитися планованими розробками з покупцями – з'явилося прототипування і почалося руйнування класичних, послідовних (каскадних) методів розробки.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

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

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

На цьому зібранні було вирішено, що Дженніфер Степлтон, тоді представляла компанію Logica, розробить архітектуру комплексного, орієнтованого на користувача методу з хорошим контролем якості для ітеративної і інкрементної розробки. Підсумкова архітектура була спроектована так, щоб бути повністю сумісна зі стандартом ISO 9000 і PRINCE2. Коли архітектура була готова (через місяць після зборів), Консорціум сформував групи для її поширення у всіх областях розробки програмного забезпечення, які включали в себе: методи та засоби управління проектом, контроль якості та тестування, методи і засоби розробки. Контрольна група, очолювана творцем архітектури і складається з глав цих груп, повинна була забезпечити розуміння методу так, як він спочатку замислювався. Незважаючи на те, що багато членів Консорціуму були прямими конкурентами, вони вільно ділилися тим, як вони вирішують проблеми, що виникають. Практика показала, що найкращий результат може бути досягнутий тільки працюючи як одне ціле. Консорціум збільшився за перший рік від декількох організацій до шістдесяти; опис методу ставало все більш і більш повним. Версія 1 була сформована в грудні 1994 року і опублікована в лютому 1995 року. Результатом став універсальний метод, що охоплює людей, процеси та інструменти. Він сформувався на основі досвіду організацій, різних за родом своєї діяльності і розмірами.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

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

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

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

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

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

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

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

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

8. Тестування інтегровано в життєвий цикл розробки.

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

Передумови для використання DSDM.

Щоб успішно використовувати DSDM, необхідно щоб був виконаний ряд передумов. По-перше, необхідно організувати взаємодію між проектною командою, майбутніми користувачами і вищим керівництвом. По-друге, повинна бути можливість поділу проекту на менші частини, що дозволить використовувати ітеративний підхід.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Два приклади проектів, для яких DSDM не дуже підходить:

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

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

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

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Madryga. Алгоритм Madryga складається із двох вкладених циклів. Зовнішній цикл повторюється вісім разів (для гарантії надійності число циклів можна збільшити) і полягає в застосуванні внутрішнього циклу до відкритого тексту. Внутрішній цикл перетворює відкритий текст у шифртекст і виконується однократно над кожним 8-бітовим блоком (байтом) відкритого тексту. Таким чином, весь відкритий текст послідовно вісім разів обробляється алгоритмом.

Ітерація внутрішнього циклу оперує з 3-байтовим вікном даних, називаним робочим кадром (рисунок 4.3). Це вікно зрушується на 1 байт за ітерацію. (При роботі з останніми 2 байтами дані покладаються циклічно замкнутими). Перші два байти робочого кадру циклічно зрушуються на змінне число позицій, а для останнього байта виконується операція XOR з декількома бітами ключа. У міру переміщення робочого кадру всі байти послідовно циклічно зрушуються й піддаються операції XOR із частинами ключа. Послідовні циклічні зрушення

перемішують результати попередніх операцій XOR і циклічного зрушення, причому на циклічне зрушення впливають результати XOR. Завдяки цьому процес у цілому оборотний.

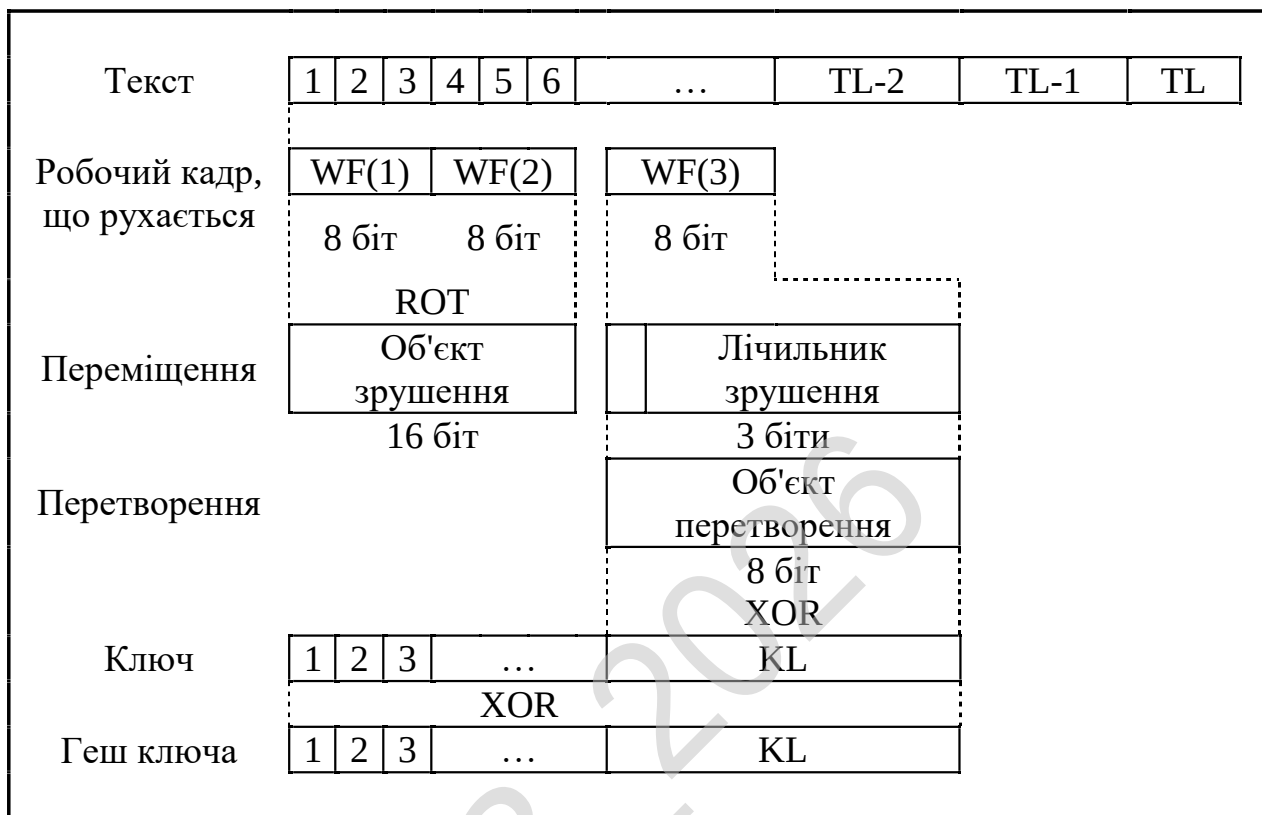


Рисунок 4.3 – Одна ітерація алгоритму Madryga

Оскільки кожний байт даних впливає на два байти ліворуч і на один байт праворуч від себе, після восьми проходів кожний байт шифртексту залежить від 16 байтів ліворуч і 8 байтів праворуч.

При шифруванні кожна ітерація внутрішнього циклу встановлює робочий кадр на передостанній байт відкритого тексту й циклічно переміщає його до третього з кінця байту відкритого тексту.

Спочатку весь ключ піддається операції XOR з випадковою константою й потім циклічно зрушується вправо на 3 біти (ключ і дані рухаються в різних напрямках, щоб мінімізувати надлишкові операції з бітами ключа).

Молодші три біти молодшого байта робочого кадру зберігаються, вони визначають циклічне зрушення інших двох байтів. Далі конкатенація двох старших байтом циклічно зрушується вліво на змінне число біт (від 0 до 7). Потім над молодшим байтом робочого кадру виконується операція XOR з молодшим байтом ключа. Нарешті робочий кадр зміщується вправо на один байт і весь процес повторюється.

Випадкова константа призначена для перетворення ключа в псевдовипадкову послідовність. Довжина константи повинна бути рівній довжині ключа. При обміні даними абоненти повинні користуватися однією й тією же константою. Для 64-бітового ключа Madryga рекомендує константу 0x0fle2d3c4b5a6978.

При розшифруванні процес повторюється у зворотному порядку. У кожній ітерації внутрішнього циклу робочий кадр установлюється на байт, третій ліворуч від останнього байта шифртексту, і циклічно зрушується у зворотному напрямку до байта, розташованого на 2 байти уліво відносно останнього байта шифртексту. 2 байти шифртексту в процесі циклічно зрушуються вправо, а ключ – уліво. Після циклічних зрушень виконується операція XOR.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система віртуалізованої серверної інфраструктури на базі HCl. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Розділу обрання групи; Розділу виведення результату роботи системи; Функції представлені у графічному вигляді.

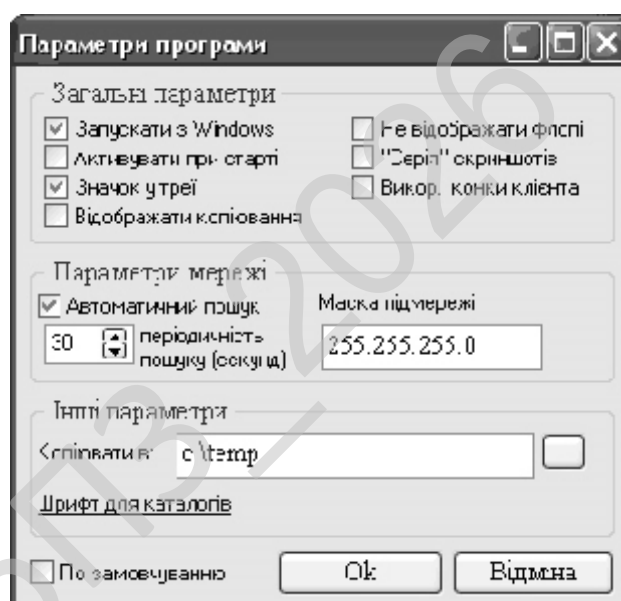


Рисунок 5.1 – Головне вікно ПЗ

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

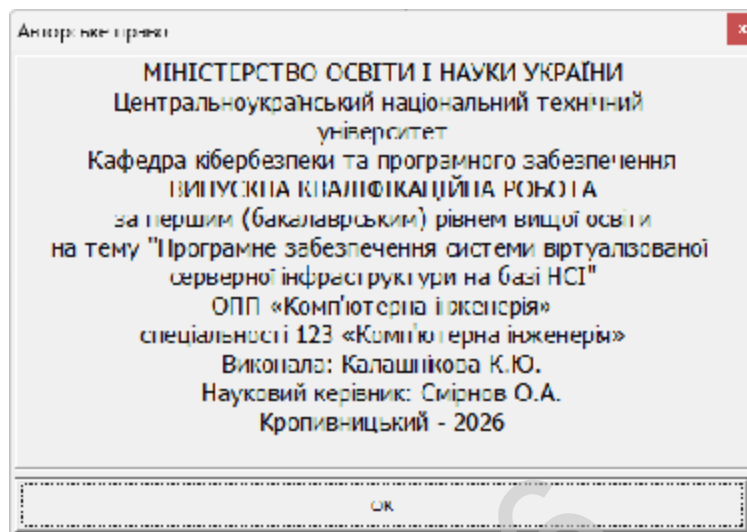


Рисунок 5.2 – Авторське право

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

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

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

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

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

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

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

Проводилась оцінка:

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

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

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

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

Ці тести демонструють:

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

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

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження

комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruvatisya), zaplativshi avtorovi певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи віртуалізованої серверної інфраструктури на базі HCl.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем віртуалізованої серверної інфраструктури на базі HCl.
- Досліджена система віртуалізованої серверної інфраструктури на базі HCl.
- На основі отриманих результатів досліджень створена програмна реалізація системи віртуалізованої серверної інфраструктури на базі HCl.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віртуалізованої серверної інфраструктури на базі HCl.

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

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи віртуалізованої серверної інфраструктури на базі HCl. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

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 Madryga.

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.

К6ПЗ-2026

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

10. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.
11. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
12. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
13. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
14. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
15. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувацького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.
16. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
17. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

					ВКРБ-123.26.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

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

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

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

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

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

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