

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
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи
гіперконвергентної інфраструктурної платформи”

КБПЗ - 2025

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

Керівник проекту
кандидат технічних наук, доцент
_____ Дреєв О.М.
« ____ » _____ 2025 р.
Рецензент _____

АНОТАЦІЯ

Драганець О.А. Дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи гіперконвергентної інфраструктурної платформи.

Метою розробки є дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи.

Об'єктом дослідження є процес гіперконвергентної інфраструктурної платформи.

Предметом дослідження є методи гіперконвергентної інфраструктурної платформи.

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

Результат роботи – програмна реалізація системи гіперконвергентної інфраструктурної платформи.

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, гіперконвергентна інфраструктурна платформа

ABSTRACT

Drahanets O.A. Research and software implementation of the hyperconverged infrastructure platform system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the hyperconverged infrastructure platform system.

The purpose of the development is the research and software implementation of the hyperconverged infrastructure platform system.

The object of the research is the process of the hyperconverged infrastructure platform.

The subject of the research is the methods of the hyperconverged infrastructure platform.

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

The result of the work is the software implementation of the hyperconverged infrastructure platform system.

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

A user-friendly user interface has been developed. Instructions for working with the software are provided.

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

The program is developed in the Python environment.

Keywords: computer engineering, hyperconverged infrastructure platform

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	36
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	57
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	57
4.2 Захист розробленого програмного забезпечення.....	71
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	74
6 НАУКОВА НОВИЗНА	81

						ВКРМ-123.25.0001.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Драганець О.А.				Дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи	Літ.	Аркуш	Аркушів
Перев.	Дресв О.М.					М	1	105
Н.контр.	Коваленко А.С.					ЦНТУ КІ-24Мз		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	82
7.1	Визначення цільової аудиторії кінцевого готового продукту	82
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	83
7.3	Вибір методу оцінки вартості ПЗ	83
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	84
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	86
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	86
7.7	Визначення ключових факторів успіху конкретного проєкту.....	87
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	88
8.1	Вступ.....	88
8.2	Аналіз умов праці на робочому місці фахівця	90
8.3	Розробка заходів з умов поліпшення охорони праці.....	93
8.4	Розрахункова частина	94
8.5	Висновки до розділу.....	96
9	ОСНОВНІ ВИСНОВКИ.....	97
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

КБПЗ-2025

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

КВ	–	коефіцієнт варіації
КЗ	–	канал зв'язку
ПС	–	програмна середа
СеМО	–	експонентна мережа масового обслуговування
СМО	–	система масового обслуговування
СПД	–	система передачі даних

КБПЗ – 2025

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Не таємниця, що для кращого використання ресурсів, у будь-яких системах, інформаційні не виключення, ресурси варто збирати в загальні пули, для того, щоб їх можна було доцільно перерозподіляти. У комп'ютерингу оптимізація в частині використання ресурсів почалася давно, ще із систем розподілу часу. Потім прийшла черга віртуалізації. У другому десятилітті ХХІ століття фокус уваги природно змістився на гіперконвергенцію й дезагрегацію, тому що на даному рівні розвитку технологій створювати ресурсні пули можна цими двома альтернативними способами. Розходження між ними полягають у відношенні до віртуалізації. Якщо залишатися на фізичному рівні, то варто діяти засобами дезагрегації. Якщо ж фізичні ресурси можна перетворити у віртуальні, то тоді переважніше засобу гіперконвергенції. Гіперконвергенція дозволяє створювати програмно-центричні інфраструктури, у яких тісно інтегровані різноманітні «апаратні ресурси» (commodity hardware), але на ринок такі інфраструктури виводяться під брендом одного вендору.

Префікс «гіпер» припускає існування й просто конвергентних систем. Спільно два типи систем утворюють підхід, що споконвічно називали інтегрованим стеком (integrated stack). Загальним для обох є те, що в масштабований стек збираються три основні компоненти – мережі, сервери й системи зберігання інформації (СЗД), і вони поєднуються засобами віртуалізації й керування.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи.

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

- Огляд існуючих систем гіперконвергентної інфраструктурної платформи.
- Дослідження системи гіперконвергентної інфраструктурної платформи.
- Програмна реалізація системи гіперконвергентної інфраструктурної платформи.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Об'єктом дослідження є процес гіперконвергентної інфраструктурної платформи.

Предметом дослідження є методи гіперконвергентної інфраструктурної платформи.

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

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

- Удосконалено метод гіперконвергентної інфраструктурної платформи.
- Розроблено вітчизняний продукт гіперконвергентної інфраструктурної платформи, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі гіперконвергентної інфраструктурної платформи.

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

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

По даним IDC, світові витрати на конвергентну інфраструктуру досягнуть \$17,8 мільярдів в 2026 році й складуть 12,8% від всіх витрат на пристрої зберігання, мережні й програмні рішення. В 2025 році ця цифра рівнялася \$4,6 мільярдам.

Конвергентна інфраструктура

Що ж таке конвергентна інфраструктура? Такий тип інфраструктури являє собою готове рішення від виробника, завдання якого – прискорити розгорнення інфраструктури. Основна ідея полягає в тому, щоб забезпечити одну крапку входу для технічної підтримки й спростити обслуговування компонентів.

Термін конвергентна інфраструктура запропонувала компанія Hewlett-Packard шість років тому. У термінології Gartner цей тип інфраструктури називається інтегрованою системою, а в Cisco Systems – системою уніфікованих обчислень (UCS).

Але яким би не була назва, за ним криється та сама ідея: об'єднання пам'яті, обчислювальних і мережних ресурсів у пул, задалегідь зконфігурований для роботи в дата-центрі. Такий підхід скорочує час розгорнення інфраструктури з декількох місяців до декількох днів.

Створення конвергентної інфраструктури в корпоративному середовищі – це щось більше, ніж проста заміна пари мережних пристроїв. У неконвергентній інфраструктурі, реалізуючи віртуальний сервер, на фізичних серверах запускається гіпервізор, керуючий віртуальними машинами, а сховище даних являє собою DAS, NAS або SAN. У конвергентній інфраструктурі сховище сполучене з фізичними серверами, а для роботи високопродуктивних застосунків і кешування даних використовується флеш-сховище.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Як приклад конвергентної інфраструктури можна привести рішення від компаній Cisco і NetApp – FlexPod. За серверну й мережну сторону відповідає компанія Cisco (мережне встаткування серії Nexus і серверне встаткування UCS), а за зберігання даних – компанія NetApp (серія FAS). У підсумку вийшла гнучка масштабована й відказостійка система.

Рішенням FlexPod користується швейцарська команда Sauber Motorsport AG, що виступає в перегонах «Формули-1». Ще в 2007 році хлопці зштовхнулися з необхідністю організації мобільного дата-центра, щоб запускати додатка для аналізу інформації про гонку, а потім передавати зібрані дані в сховища стаціонарних ЦОД.

Компанія розгорнула двухвузловий кластер, що складається з NetApp FAS2040 і NetApp SyncMirror для реплікації даних. Мобільний центр даних містить 8 блейд-серверів Cisco UCS і світч Cisco Nexus.

Отримані під час перегони дані за допомогою NetApp SnapMirror реплікуються на MetroCluster у дата-центрах міста Хінвіль. Середня затримка при передачі даних з гоночної траси в головний офіс становить близько 15 хвилин. За допомогою платформи FlexPod відбувається збір даних про боліди. Інформація про використання палива, температури, двигунах надходить із сотень сенсорів. Команда використовує телеметричні показники разом з даними про стан траси, запускаючи симуляції в FlexPod, щоб зрівняти, як віртуальна модель співвідноситься з поточними налаштуваннями автомобіля. Це допомагає інженерам команди реагувати на зміни в ході перегони.

1.2 Область застосування

Гіперконвергентна інфраструктура

Гіперконвергентні системи переводять концепцію конвергентності на новий рівень. Важливою відмінністю між двома технологіями є те, що в конвергентній інфраструктурі кожний компонент у будівельному блоці є

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

дискретним і може використовуватися окремо. Що стосується гіперконвергентної інфраструктури, те це програмно-визначаєма технологія, тому всі компоненти інтегровані.

Гіперконвергентні рішення виділяються поліпшеннями на рівні програмного контролера, що дає можливість легко їх масштабувати. Щоб збільшити ємність і продуктивність, потрібно додати новий блок. Замість нарощування потужності за рахунок збільшення числа дисків, кількості пам'яті або процесорів, продуктивність збільшується за рахунок додавання більшого числа модулів.

Таким чином, гіперконвергентна інфраструктура – це інфраструктура, у якій обчислювальні потужності, сховища, сервери, мережі поєднуються в одне ціле за допомогою програмних засобів, а керування ними відбувається через загальну консоль адміністрування. Із цієї причини замість команди ІТ-фахівців для керування сховищами даних і серверним устаткуванням часом досить одного системного адміністратора.

Серед мінусів гіперконвергентних систем варто відзначити неможливість гранулярного апгрейду. Збільшення обсягу сховища й підвищення продуктивності є критично важливими пунктами для будь-якої компанії, але, якщо місце на СЗД-кластері підходить до кінця, а обчислювальних ресурсів цілком достатньо, вам все-таки прийде збільшити загальну обчислювальну потужність, додавши новий модуль.

Великим постачальником гіперконвергентної інфраструктури є компанія VMware. Комплексним рішенням VMware EVO:RAIL користується японський банк Fukuoka Hibiki Shinkin. Атсуши Йошида (Atsushi Yoshida), представник банку, споконвічно звернув увагу на EVO:RAIL у силу гнучких можливостей інтеграції з наявними серверами, системами зберігання даних, гіпервізором і віртуальним програмним забезпеченням.

Використання гіперконвергентного рішення дозволило банку знизити експлуатаційні витрати до 10% і зменшити показник TCO (total cost of ownership)

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

на 40%. Увесь час реалізації, з урахуванням створення віртуальних машин, зайняло 15 хвилин.

Як у випадку із традиційними інфраструктурами, вартість гіперконвергентної інфраструктури може варіюватися залежно від використовуваного гіпервізору. Інфраструктура на VMware vSphere або Microsoft Hyper-V може коштувати досить дорого, а от Nutanix підтримує «опенсорсний» KVM.

Для тих, хто шукає спосіб використовувати вже існуюче встаткування для побудови гіперконвергентної інфраструктури, кілька компаній пропонують програмні технології, що підвищують продуктивність існуючої інфраструктури. Однією з таких компаній є Atlantis of Mountain View, чиє рішення перетворює DAS у масив, збільшуючи число VM, що працюють зі сховищем.

Підводячи підсумки, можна сказати, що конвергентні й гіперконвергентні рішення, пропоновані різними компаніями, допомагають заощаджувати час на впровадженні й технічній підтримці інфраструктури; пропадає необхідність у великій кількості адміністраторів, що здійснюють дану підтримку. Компанія, що впровадила такі рішення, підвищує ефективність IT-відділу й свого бізнесу.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Ринок IT-інфраструктури стрімко міняється. За станом на початок 2026 року гіперконвергентні системи вже замінили звичні сервери й системи зберігання в 20% українських компаній. За наступні три роки 10% бюджетів на сервери й СЗД дістануться виробникам гіперконвергентних рішень. Такі результати дослідження TAdviser «Актуальні тенденції ринку гіперконвергентної інфраструктури».

Швидкий ріст

Метою дослідження TAdviser було визначити рівень поширення технологій гіперконвергентності серед українського бізнесу й держструктур, а також оцінити потенціал попиту на подібні рішення в Україні в короткостроковому періоді. Як показало дослідження, український ринок гіперконвергентних інфраструктур (НСІ) перебуває в стадії росту. Його обсяг в 2018 році, по оцінці TAdviser, склав близько 1 млрд гривень. У той же час IDC прогнозує ріст глобального ринку гіперконвергентних інфраструктур на 150% у рік, а Gartner очікує, що до 2020 року 20% бізнес-критичних застосунків будуть перенесені на гіперконвергентні платформи.

Ринок НСІ (як глобально, так і в Україні) буде рости за рахунок міграції із традиційних платформ. З погляду динаміки обсягу ринку IT у цілому можна буде говорити про перерозподіл витрат на користь гіперконвергентних рішень по статті «устаткування». Уже в найближчі три роки НСІ може «від'їсти» до 10% витрат на сервери й СЗД.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Гіперконвергентною інфраструктурою в даному дослідженні вважаються горизонтально масштабовані апаратно-програмні комплекси на базі x86-вузлів з єдиним обчислювальним середовищем і середовищем зберігання, що не вимагають для функціонування виділених систем зберігання даних і в які всі необхідні інфраструктурні функції (такі як резервування, реплікація, висока доступність, відновлення) реалізуються на програмному рівні.

До останнього часу сама концепція НСІ сприймалася замовниками як недостатньо зріла. Її більше активне просування сповільнювали як усе ще властива великому українському бізнесу консервативність, так і недостатній рівень поінформованості про існуючі пропозиції і їхні можливості. Навпроти, головні стимули ініціатив на поле НСІ сьогодні – інтерес до міграції в хмару, уже досягнута зрілість віртуалізації, а також явна невідповідність можливостей застаріваючої інфраструктури новим бізнес-завданням. Основними замовниками таких рішень стають телекому-компанії, банки, а також підприємства нафтогазової й промислової галузей економіки.

Кількість гравців на цьому ринку росте – в основному це великі інфраструктурні вендори. У числі лідерів – Dell EMC, HPE, Nutanix. Також аналітичні агентства Gartner і Forrester виділяють Cisco, Huawei, Hitachi, Lenovo і NetApp. За даними опитування 451 Research, 47,5% представників американського великого бізнесу відзначили, що використовують НСІ-рішення VMware, 41% – Dell EMC, 24% – Cisco, 17% – HPE і 15% – Nutanix.

Нові вимоги ринку

Гіперконвергентні рішення орієнтовані на швидко розвивається або динамічно, що трансформується бізнес, забезпечуючи йому можливість швидко «включати» і «виключати» необхідні ІТ-ресурси під мінливі вимоги й нарощуючи продуктивність до необхідного рівня. Це дозволяє більш оперативно впроваджувати ідеї в практичну реалізацію, а значить – істотно прискорювати ТТМ (Time-to-market), щоб швидше виводити на ринок нові продукти. Саме цей аспект сьогодні є ключовим для більшості цифровізуючихся галузей. До інших

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

значимих ефектів від гіперконвергенції відносять можливість оптимізації витрат і спрощення модернізації інфраструктур. Зокрема, засоби віртуалізації дозволяють вирішувати проблему неефективного використання серверних ресурсів, а перехід до централізованої інфраструктури і єдиних інтерфейсів підвищує ефективність керування обчислювальними потужностями.

За даними опитування TAdviser, близько 50% респондентів, що представляють великий бізнес і держсектор, розглядають сьогодні можливість відновлення своїх ІТ-інфраструктур. Стимулює до цього в першу чергу проблема продуктивності, з якої всі частіше зіштовхуються в цих організаціях (80%), у тому числі, у виді старіння раніше закупленого встаткування (75%). Нові вимоги до продуктивності диктують також зростаючі потреби в просунутій аналітиці, у побудові нових бізнес-процесів і цифрових платформ (наприклад, інтернету речей, додатка якого орієнтовані на роботу в масштабованому й гнучкому хмарному середовищі).

Готовність до змін

21% опитаних організацій уже використовує гіперконвергентні системи, ще 34% – знайомо з такими рішеннями й готово розглянути їхнє впровадження. Сам термін уже не можна назвати екзотичним, але загальне подання замовників про пов'язаній з ним концепції зводиться переважно до віртуалізації й частки хмарі.

Більше половини з опитаних TAdviser вважають НСІ сучасним архітектурним підходом. У числі головних переваг від використання гіперконвергентних рішень біля половини респондентів указує простоту в обслуговуванні (48%), легку масштабованість (46%), а також більше низьку вартість, у порівнянні із традиційним підходом до побудови або модернізації інфраструктури (39%). На думку респондентів TAdviser, використання НСІ доцільно розглядати під СУБД (53%), аналітичні системи (45%), а також сервери застосунків (40%).

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Більше 70% опитаних підтверджують готовність використовувати українські ІТ-продукти, або вже їх впровадили. Серед основних факторів, що стимулюють інтерес до українських продуктів, замовники відзначають вартісні переваги в порівнянні із закордонними рішеннями, довіра в питаннях безпеки до вітчизняних розробок у порівнянні із західними.

Гіперконвергентні інфраструктури (НСІ) стають одними з найбільш перспективних технологій для створення приватних і гібридних хмар. Ідею НСІ запропонували стартапи, які раніше інших зрозуміли можливість радикальних змін і що встигли зайняти лідируючі позиції в цьому сегменті ринку. Однак і галузеві великовагові поїзди, що як звичайно затрималися на старті, але володіють достатнім науковим і технічним потенціалом, роблять все можливе, щоб зайняти належне їм положення.

Загальна ситуація така: незалежно від того, під яким брендом виробляється та або інша конкурентна НСІ, вона складається із чотирьох взаємозалежних компонентів – сервери, СЗД, мережа й віртуалізація СЗД із метою створення загальних пулів (Storage Pool). Щодо цього, мабуть, єдине виключення являє собою трикомпонентний підхід до утворення НСІ на базі IntelXeon Scalable із чип-сетом Intel C620 і СЗД Optane SSD.

Як постачальник НСІ за рідкісним винятком виступають компанії, що забезпечують функціонал, а саме два перших з перерахованих вище чотирьох компонентів. Вони можуть самі робити сервери й/або СЗД, або ж можуть здобувати їх, практично не маючи власного виробництва, тобто бути, як говорять, фаблесс.

Що ж стосується двох інших компонентів (віртуалізаційне ПЗ й мережне встаткування), які можна назвати системоутворюючими, те тут картина зовсім інша. У цих сегментах є кілька безумовних лідерів і саме вони задають правила гри.

Найбільш популярним і із системної точки зору установлюючим стандарт де-факто став дует Microsoft и Mellanox. Microsoft надає свій засіб для створення пулів зберігання Storage Spaces Direct (S2D), що входить в Server 2025, а

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Mellanox – повний стек мережних технологій. Цьому сполученню віддають свої перевага такі універсальні вендори як HPE, Dell-EMC, Supermicro і спеціалізуються на HCI, наприклад, dataON.

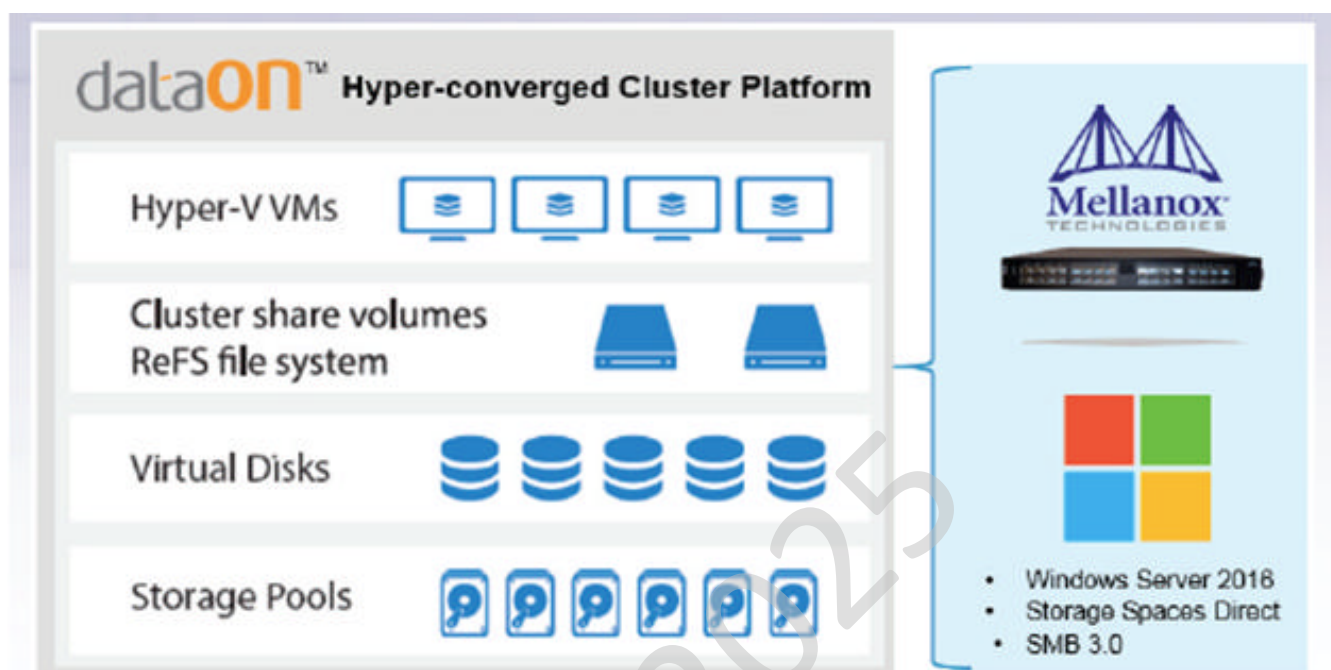


Рисунок 2.1 – Приклад використання продуктів Microsoft і Mellanox в HCI від компанії dataON

Технології Microsoft

Програмно-визначаємий підхід до СЗД (Software-Defined Storage, SDS) і технологія віртуалізації дискового простору вперше з'явилися в Windows Server 2012 під ім'ям Storage Spaces. В Windows Server 2025 увійшов оновлений і більше успішний варіант Storage Spaces Direct (S2D). S2D здатний збирати в загальні пули (репозиторії) і, як треба з назви, забезпечувати прямий доступ до просторів на дисках SATA, SAS і NVMe, розміщених усередині серверів.

Суть концепції дискових пулів полягає в тому, що окремі фізичні диски сервера поєднуються в один або кілька пулів, на базі яких створюються томи з різними рівнями продуктивності й відказостійкості. В одному пулі можуть

перебувати диски, що відрізняються по обсязі, продуктивності й інтерфейсу підключення. Віртуальні диски служать логічними томами. Вони можуть бути:

- простими (simple) аналогами RAID-0, забезпечуючи оптимальне використання ємності дисків. Будучи самими продуктивними, вони не припускає ніякий відказостійкості;

- дзеркальними (mirror) аналогами RAID-10 або 1E для непарного числа дисків. Такий віртуальний диск захищений від втрати одного або двох дисків відповідно;

- с контролем парності (parity) аналогами RAID-5 і RAID-6 ВД, допускаючи втрату одного або двох дисків відповідно.

S2D підтримує:

- двохярусне зберігання даних (Storage tiers), тобто розміщення даних на віртуальних дисках Storage Spaces на SSD і HDD. Для цього при створенні томів (віртуальних дисків) вказується комбінація обсягів SSD і HDD на кожному з ярусів, файл можна прив'язати до SSD-Ярусу для забезпечення гарантованої продуктивності;

- кеш для запису (write-back cache). Для компенсації низкою HDD продуктивності частина SSD у пулі використовується для кешування запису, при цьому частина, що залишилася, ємності SSD може бути використана для ярусного зберігання.

Кількісні характеристики S2D:

- До 240 дисків в одному, пулів може бути кілька.
- До 80-ти дисків у пулі, до 4-х пулів на кластер.
- До 64-х віртуальних дисків в одному пулі.
- Загальна ємність пула – до 480 Тб.

S2D реалізується у двох варіантах, по п'яти або шестирівневої моделі. На рисунку нижче показана п'ятірівнева модель що включає:

- Рівень серверів, зв'язаних по мережному протоколі прикладного рівня SMB (Server Message Block), службовцеві для вилученого доступу до файлів, принтерам і іншим мережним ресурсам і для міжпроцесної взаємодії. У першій

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

редакції він називався Єдина файлова система інтернету (Common Internet File System, CIFS) і був розроблений IBM, Microsoft, Intel і 3Com в 1980-х роках. SMB був реалізований через NetBIOS (поверх NBF, IPX/SPX або NetBIOS over TCP/IP) і використовувався в мережах MS-NET і LAN Manager для DOS, а також в Windows for Workgroups.

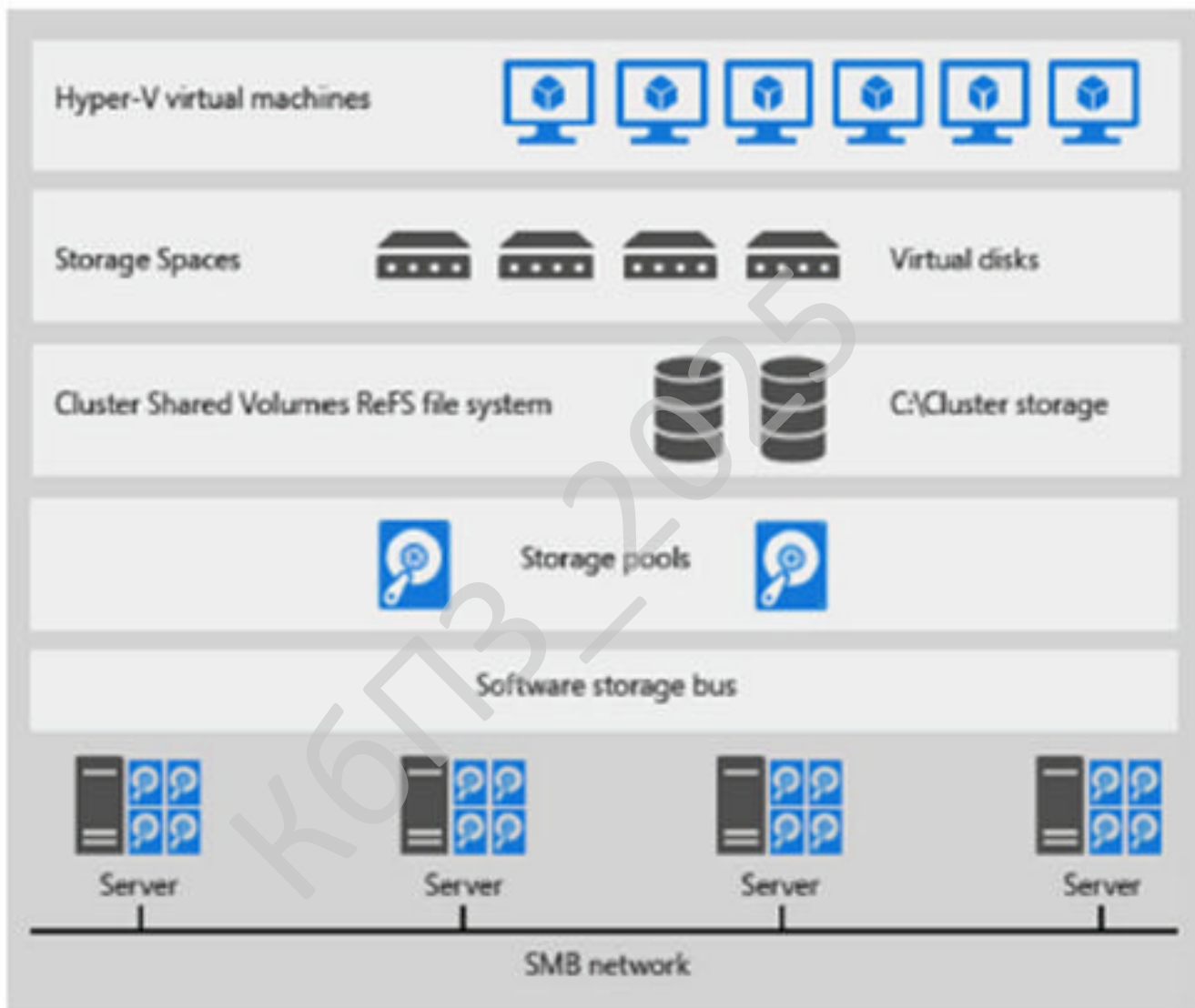


Рисунок 2.2 – Стэк технологій Storage Spaces Direct

– Шина Software Storage Bus охоплює всі сервери, що становлять кластер. При використанні Software Storage Bus кожний сервер може «бачити» всі диски

на всіх серверах у кластері, забезпечуючи з'єднання в мережі з повнозв'язною топологією.

– Пул зберігання (Storage pool) – набір фізичних дисків. У нього можуть входити диски, що відрізняються по обсязі, продуктивності й інтерфейсу підключення.

– ReFS (Resilient file system) – як і SMB, річ не нова. Ця безвідмовна файлова система, своїми корінням іде в OS/2 і IBM PS2.

– Локальні дискові простори (Storage Spaces) – віртуальні дискові томи.

– Віртуальні машини. Цей шостий рівень присутній, якщо реалізується гіперконвергентний варіант архітектури. Якщо ж обмежуються конвергентною архітектурою або, як її іноді називають дезагрегацією, то використовується п'ять нижніх рівнів. Вибір між дезагрегацією і гіперконвергенцією визначається вимогами користувача. Перший варіант переважніше для менших по масштабах конфігурацій, а другий для більших.

Технології Mellanox

На технології Microsoft доводиться програмна складова описаної вище архітектури, а безпосередньо обмін даними забезпечує Mellanox. Компанія розташовує необхідним для цього повним стеком технологій. Перевага Mellanox полягає в тому, що пропоновані цією компанією продукти відповідають рекомендації Microsoft щодо використання прямого доступу на згадку RDMA зі швидкістю передачі даних між вузлами кластера не менш 10 Гб/сек.

Локальна технологія прямого доступу до пам'яті Direct Memory Access (DMA), тобто звернення до пам'яті без порушення роботи CPU, відома давно. Вона широко використовується для підключення периферійних пристроїв. Але з появою високошвидкісних мережних технологій спочатку відкрилася можливість для дистанційного прямого доступу RDMA (Remote Direct Memory Access), а потім і RoCE (RDMA over Converged Ethernet), тобто по вдосконаленій мережі Converged Enhanced Ethernet (CEE), що відрізняється більше високою

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

пропускною здатністю й меншими затримками. Застосування RDMA почалося з високопродуктивного комп'ютеринга HPC і поширилося на HCl.

Технологія реалізується в мережних контролерах Mellanox, починаючи з Connect-3 Pro і далі. Нове сімейство мережних карт Connect-4 дозволяє спростити впровадження RoCE і робить технологію доступною для хмарних систем і СЗД.

Connect-4 Lx EN із пропускною здатністю 1/10/25/40/50 Гбіт/с спроектований для рішення завдань віртуалізованих інфраструктур і відрізняється найвищою у своєму класі продуктивністю для різних застосунків і ринків. У цих адаптерах справжня апаратна ізоляцію уведення-виводу сполучається з унікальною масштабованістю й ефективністю. Connect-4 Lx EN підтримує швидкості 1, 10, 25, 40 і 50 Gb, забезпечує затримку менш 1 мікросекунди й швидкість передачі повідомлень 75 млн пакетів у секунду.

Великий пакет пропозицій, що включає коммутатори Mellanox на платформі Open Ethernet, складається із двох сімейств Switch-2 і Spectrum. Сімейство Open Ethernet містить у собі різноманітне портфоліо комутаторів з форм-фактором 1RU, кількістю портів від 12 до 64 і інтерфейсами від 1 до 100 Гбіт/с, що дозволяють спроектувати спеціалізований центр обробки даних будь-якого розміру з будь-якою величиною блокування. У лінійці компанії Mellanox присутні високопродуктивні системи комутаторів з архітектурою Open Ethernet і інтерфейсами 10/25/40/50/56/100 Гбіт/с, які забезпечують максимальну ефективність роботи центрів обробки даних, рішень для хмарних обчислень, зберігання даних, високоефективних обчислень і Web 2.0 у будь-яких масштабах.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

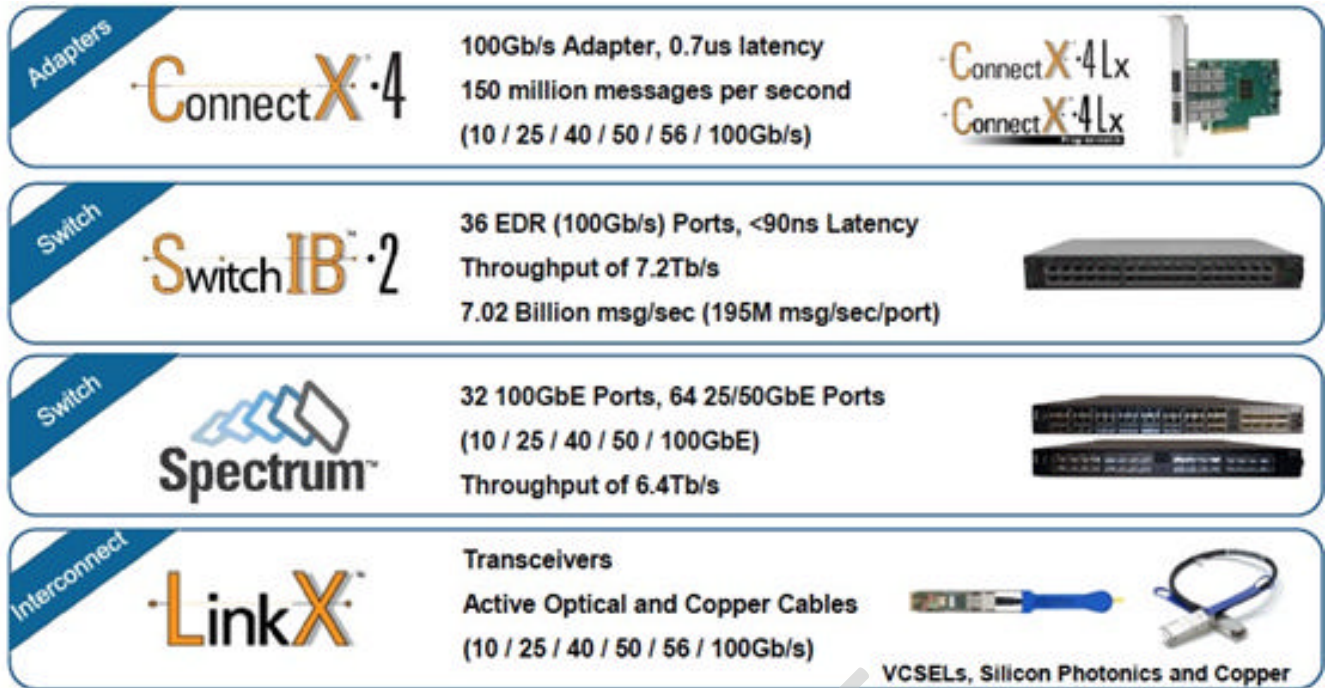


Рисунок 2.3 – Стек технологій Mellanox

Mellanox пропонує широкий вибір кабелів різної довжини, пропускної здатності й типів носіїв. Для фізичної комутації HCI використовуються сертифіковані кабелі й трансівери Mellanox Link, вони відповідають всім вимогам стандартів IEEE, а також пройшли 100 % випробувань для забезпечення оптимальної цілісності сигналу й наскрізної продуктивності. Кабелі Mellanox підходять для рішень на базі Ethernet, InfiniBand, хмарної інфраструктури й Web 2,0, а також підключень сервер-стійковий комутатор, сервер-сЗД, комутатор-комутатор.

Майбутнє HCI – Microsoft Azure Stack

Azure Stack – це дуже потужна заявка Microsoft на місце провідного постачальника рішень для гібридних хмар. Пропонована стратегія припускає відмову від самодіяльності при створенні власної платформи й для часток, і гібридних хмар. Замість цього пропонуються готові HCI, що поставляються або самої Microsoft, або обмеженим колом партнерів OEM (HPE, Dell, Lenovo і деякі інші). Такі системи будуть повністю сумісні із хмарою Microsoft Azure, що

працює на RoCE. Для мереж у них буде застосовуватися встаткування від Mellanox, у тому числі мережні адаптери Connect, комутатори Spectrum, трансівери й кабелі (мідні й оптичні) Link.

Використання перевірених HSI-рішень забезпечує безшовне узгодження між глобальною хмарою Azure і гібридною хмарою Azure Stack. А той факт, що апаратне забезпечення буде надходити від Microsoft і її партнерів, гарантує всі його необхідні експлуатаційні показники.

Потенційний успіх Azure Stack дає деяким спостерігачам говорити про майбутні зміни в гартнеровському квадранті.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python <https://www.python.org/> і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для налаштовуваних програм.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи гіперконвергентної інфраструктурної платформи.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

відноситися й керування інфраструктурою віртуальних робочих столів, залежно від обраної конфігурації й виробника.

Апаратні засоби, що йдуть у комплекті конвергентної інфраструктури, попередньо налаштовані на підтримку будь-яких цільових завдань: інфраструктури віртуальних робочих столів, бази даних, спеціальні додатки й так далі. Але фактично у вас немає достатньої волі для зміни обраної конфігурації.

Незалежно від обраного способу розширення середовища інфраструктури віртуальних робочих столів, ви повинні розуміти, що наступне вертикальне масштабування буде витратним і зажадає багато часу. Додавання окремих компонентів перетворюється стає складним і позбавляє вас багатьох переваг конвергентної інфраструктури. Додавання робочих машин і розширення ємності зберігання в корпоративній інфраструктурі може бути таким же дорогим, що говорить про необхідність правильного планування будь-якого розгортання інфраструктури віртуальних робочих столів.

З іншого боку, усе компоненти конвергентної інфраструктури можуть працювати протягом тривалого часу. Приміром, комплектний сервер такої інфраструктури працює добре навіть без інших компонентів інфраструктури.

Розвиток ІТ-Технологій не стоїть на місці, у світло виходять нові більше ефективні рішення, і одним з таких нововведень є конвергентні інфраструктури.

Converged Infrastructure (CI), або конвергентні інфраструктури, – це відроджені мейнфрейми, технологія CI насамперед дозволяє зібрати разом обчислювальні потужності роз'єднаних серверів, технології зберігання даних, що дозволяє забезпечувати більше ефективну динамічність систем. Поряд з розвитком цих технологій з'явилася й нечисленна категорія інших машин – analytic appliance, які дозволяють обробляти більші масиви даних буквально в режимі реального часу.

Що ж таке CI – конвергентна інфраструктура?

– CI – насамперед – це сукупність взаємодоповнюючих компонентів – модулів зберігання даних, серверів, програмного забезпечення, мережного встаткування, керування, а так само засобів автоматизації.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

самостійного продукту. Існує 14 конфігурацій, що підтримують хмарні додатки, у кожній з яких відбувається різне сполучення компонентів – керуючого ПЗ, гіпервізори, стійкові сервери, комутатори й системи зберігання.

Стекові інфраструктури

До таких інфраструктур можна віднести Dell Active System (раніше називалися vStart) і IBM PureSystems.

Dell Active System

Існує чотири версії Dell Active System, у кожній з яких зібрані в одну стійку системи зберігання, сервери, що управляє ПЗ. Кожна версія має індекс, що вказує число віртуальних машин – Active System 50, Active System 200, Active System 800 і Active System 1000. Установка цих систем досить проста, за словами розроблювачів, такі системи можуть являти собою подоба корпоративного iPod, у якому є тільки саме необхідне, тому що система призначена для простих користувачів, які не повинні зазнавати труднощів про відновленні ПЗ або моніторингу пристроїв. Саме рішення виконується в заводських умовах, користувач лише підключає його до мереж – електричної й інформаційної. Dell Active System складається із серверів PowerEdge, систем керування Dell Management, комутуючих пристроїв PowerConnect і пристроїв зберігання EqualLogic.

Всі системи оснащені передвстановленим гіпервізором, причому, якщо назва кінчається на m, те, виходить, система оснащена Microsoft Hyper-V, а v – VMware ESXi. Рішення vStart комплектуються Windows Server 2008 R2 Datacenter Edition, що надає більші можливості віртуалізації й використання Dell OpenManage, для якого необхідна наявність спеціального процесора, що відслідковує системи зберігання й сервери. Моделі Dell Active System комплектуються двомісячною ліцензією ESXi, а так само консоллю vCenter Server.

Половина стійки займається версією Active System 50, де розташовані сервера PowerEdge R630 у кількості двох штук і пам'яттю до 64Гб у кожному, а

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

так само комутатором на 24 порту, що підключається до мережі зберігання, і, крім того, дисковим масивом EqualLogic PS6100X обсягом 28,8 Тб.

Версія Active System 100 оснащена одним сервером PowerEdge R630, трьома Dell R730 і сховищем PS6000XV, а версія ctive System 200 уже має шість серверів R730 і два накопичувачі, крім того, обидві ці версії оснащені накопичувачем типу PowerConnect 6248.

IBM PureSystems

Розробка цієї системи велася ще з 2008 року, однак, про вихід продукту заявили лише у квітні 2012. Поки ще не можна повністю оцінити готове рішення, компанія IBM представляє лише перші наробітки в цій сфері. Фахівці затверджують, що поява PureSystems порівнянна з появою в 1964 році мейнфрейма System/360. Поряд із цим, IBM знову здобуває статус розроблювача закінчених рішень. Багато хто затверджують, що з появою PureSystems неминуче відбудеться поступовий відхід блейд-серверів.

Рішення PureSystems – це прорив, у ньому все ново – починаючи від самого виконання й закінчуючи системою керування. Загалом кажучи, PureSystems – це сімейство, що поки що містить у собі два члени – PureFlex System, а, крім того, PureApplication System. Перше рішення має три версії.

IBM PureSystems – прогрес даного рішення в наявності, навіть у конструктивних особливостях. Розроблювачі відмовилися від стандартного 1U форм-фактора, розробивши нове поняття «шасі», що може бути заповнено будь-яким устаткуванням, створеним у нових універсальних форм-факторах. Подібне рішення – це можливість для інтеграції в одну інфраструктуру всіляких систем – мережних і обчислювальних ресурсів, пристроїв зберігання й т.д. конструктивно система виконана у вигляді шасі 10U, причому, одна стійка може вмістити в себе 4 таких шасі. Подібна конструктивна реалізація доводить, що IBM зволіла піти від серверів-лез.

PureFlex System створена як інфраструктурна основа приватних хмар, складається із двох основних компонентів – це обчислювальні вузли,

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Lenovo IBM Flex System здатна містити 4 різних обчислювальні моделі. Приміром, система Flex System x240, а так само IBM Flex p24L містить два сокета, призначених для ядер Intel Xeon, система IBM Flex p460 оснащена чотирма сокетами для процесорних рішень IBM Power7, а Flex System p260 – два сокета для того ж Power7, однак, всі моделі можуть мати до десяти різновидів, які розрізняються кількостями ядер, параметрами адаптерів і т.д.

За гнучкість уведення-виводу відповідають до дев'яти контролерів, але в системі PureFlex відсутні убудовані системи зберігання інформації, за неї відповідають накопичувачі Storwize V7000.

Мабуть, навіть важко представити, які гігантські можливості надає подібний універсальний конструктор, тому для полегшення вибору компанія IBM виділяє три основних комплектації. Таким чином, для малого або середнього бізнесу призначаються System Express, System Standard – це підтримка стандартних корпоративних пропозицій, а System Enterprise – система, призначена для СУБД із більшим обсягом інформації й з підвищеною вимогою надійності.

PureApplication System

PureApplication System не сильно відрізняється від вищезгаданої системи PureFlex, однак, вона характерна не занадто розвинутою мережною підсистемою, але більшою кількістю процесорів, крім того, рішення має убудовану систему зберігання. Рішення PureApplication дозволяє встановлювати від 96 до 608 ядер, оперативна ж пам'ять варіюється від 1,5 Тб до 9,7 Тб, сховище даних має обсяг 6,4 Тб на твердотільних дисках і 48 Тб звичайних жорстких дисків.

Існують так само платформи, призначені для певних класів застосунків. В 2008 році у світло вийшла платформа Exadata – Oracle Database Machine, що була створена разом з HP. Однак, уже через два роки була розроблена машина сполучні ПЗ – Exalogic Elastic Cloud. Обоє рішення призначені для створення хмарних сервісів, характерні горизонтальною масштабованістю, а так само забезпеченням швидкої підтримай і розгортання застосунків.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Exalogic, в основному, призначена для взаємодії з Oracle Fusion Middleware, Oracle Fusion Applications, а так само Enterprise Java, однак, поряд із цим дозволяє використовувати й додатка, адаптовані до Weblogic. Дане рішення являє собою повністю зконфігуровану закінчену систему, що містить у собі сервери, модулі зберігання інформації, комутаційні компоненти. Існує три базові конфігурації – Full Rack, Half Rack і Quarter Rack, у кожній з яких є в наявності засобу резервування, необхідні мережні елементи, а так само кластеризована система даних. Конфігурація стійки дозволяє мати від 8 до 30 модулів обчислення як x86, так і x64. Цікаво, що обчислювальні модулі автономні й мають власне окреме живлення, а так само по двох накопичувача SSD, на яких утримуються діагностична інформація, а так само образ операційної системи. Система Exalogic заснована на мережі InfiniBand Fabric, що може зв'язувати компоненти, одержуючи, у підсумку, єдиний мейнфрейм. Подібне середовище розроблене для корпоративних Java-застосунків, крім того, існує друга версія цієї системи, у якій замість SSD твердотільних накопичувачів присутні звичайні магнітні жорсткі диски, що відповідають вимогам СУБД.

HP CloudSystem Matrix

А от рішення HP CloudSystem Matrix засноване на архітектурі BladeSystem. Модель складається із сукупності мережних і обчислювальних елементів, модулів зберігання інформації, елементів живлення й системи охолодження. На даний момент відомі дві моделі – це BladeSystem Matrix, зібраний на RISC-процесорах, або на архітектурі x86. Базовий комплект цього рішення являє собою набір двох модулів комутування Virtual Connect Flex-10, сервер управління HP ProLiant DL360, а так само ПЗ Matrix Operating Environment, а рішення з ОС HP-UX характерно наявністю ліцензії HP-UX 11i v3 VSE-OE Integrity. Щоб забезпечити постійну доступність, а так само відновлення у випадку яких-небудь збоїв на платформу Integrity додається компонент HP Serviceguard.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Інфраструктура розроблена таким чином, що дозволяє планувати навантаження й виділення ресурсів системи, а так само вносити які-небудь необхідні зміни. У системі відбувається регулярне фіксування таких показників, як потужність, завантаження мережних ресурсів і процесорів. Великий спектр рішень HP BladeSystem Matrix, убудованих у систему може захистити й автоматично відновити робочий стан компонентів, а так само ліквідувати наслідку збоїв. За допомогою цієї системи віртуальні й фізичні сервера вертаються в робочий стан за секунди, а система Virtual Connect дозволяє швидко створювати локальні мережі, призначені для зберігання інформації всієї інфраструктури. подібна технологія дозволяє значно спростити зв'язність сервера й збільшити пропускну здатність мережі.

Технологія Matrix Orchestration Environment дозволяє значно спростити процес створення інфраструктури для конкретного додатка, а так само подальша його підтримка. У ході планування Matrix Orchestration Environment розробляє шаблон, у якому втримується опис серверів, а так само їхніх з'єднань між собою, після чого шаблон уже використовується для розробки конкретної конфігурації при створенні певної інфраструктури в Matrix.

Існує так само досить численна категорія інфраструктур, поява якої було обумовлено наявністю проблем великого обсягу даних. Подібні системи досить різноманітні, однак, їх поєднує повсюдне використання архітектур на основі масового паралелізму.

Подібні системи розробляються з 1984 року, уперше цією проблемою всерйоз зацікавилася фірма Teradata, її сьогоdnішній продукт – це Active Enterprise Data Warehouse, що характерно наявністю власного зв'язку за назвою Vynet. Крім того, у цей час компанія Netezza пропонує власне рішення за назвою Performance Server, де відбувається сполучення стандартних лез зі спеціальними Snippet Processing Unit.

А от деякі компанії можуть запропонувати винятково програмні рішення даної системи, засновані на MPP-конфігураціях, як апаратна частина які

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

використовую стандартні сервери. Універсальні такі системи тим, що не мають особливої прив'язки до тої або іншої апаратної конфігурації, у їхній основі лежить технологія MapReduce. Як приклад, може виступати рішення Vertica Analytics Platform, однак, система не виявляє особливих вимог до апаратної частини, єдине побажання розроблювачів – це двоядерні процесори – одне ядро призначене для ОС, а друге – для роботи з даними. Все інше, що необхідно для створення самої СУБД, уже закладено в системі.

Oracle Exalytics характерна наявністю 1 Терабайт оперативної пам'яті, а, крім того, 40 процесорів, а так само системою TimesTen In-Memory Database, що використовує алгоритм збереження даних, використовуваних найчастіше.

Конкурентами Exalogic поки що є IBM Smart Analytics Systems, а так само EMC Greenplum, однак, на відміну від цих систем, Exalytics здатний взаємодіяти тільки з Exadata, що служить джерелом даних Adaptive In-Memory Cache.

І останнє рішення в сфері конвергентних інфраструктур – це щира конвергентна інфраструктура, подібне рішення вийшло в результаті спільної роботи трьох корпорацій -Cisco, Dell EMC, VMware, а так що само примкнули Intel. У результаті їхньої діяльності світло побачила спільна організація за назвою Virtual Computing Environment. Дана фірма пропонує набір продуктів Vblock, характерних наявністю наступних компонентів – системи віртуалізації, такі як VMware vCenter Server, VMware vSphere, комутаторів Nexus і Cisco, системи обчислення – Cisco Unified Computing System, а так само технологій зберігання інформації EMC Symmetrix VMAX, систем безпеки RSA enVision, а так само апаратних елементів – кабелів, стійок, ІБЖ і т.д.

Подібне рішення досить непросте, тому його остаточне складання має деякі складності. У зв'язку із цим три корпорації об'єднали виробництво й поставку готових рішень. Така єдність значно спрощує процес створення й розгортання інфраструктур, тому що кожна модель перевірена й оптимізована для більше ефективного використання наявних ресурсів. Існує кілька версій Vblock – 0, 1, Series 300 і 700, які відрізняються розміру й технологіями

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

зберігання даних. Початкове рішення – це версія Vblock 0, що сконструйована на накопичувачі Celerra, а саме потужне рішення – це версія Vblock 700 на системі зберігання Symmetrix. Крім того, рішення середнього рівня, такі як Vblock 300, створені на основі накопичувача VNX. EMC Atmos – це система об'єктного зберігання, призначена для хмарних обчислень, що підтримує обсяги даних до петабайта.

Загалом, розвиток конвергентних інфраструктур – це прорив у сфері ІТ-Технологій, що приведе до відсутності необхідності безлічі фахівців, що працюють у сфері певних технологій. Крім того, безліч підприємств зможе підібрати необхідні для себе рішення, налаштовані для рішення тих або інших виробничих завдань.

Що таке гіперконвергентна інфраструктура?

Гіперконвергентна інфраструктура була побудована на базі конвергентної інфраструктури й поданні про програмно-визначаємий ЦОД. Вона поєднує все компоненти звичного нам ЦОД в одній системі. Всі чотири ключових компоненти конвергентної інфраструктури на місці, але іноді в неї також входять додаткові компоненти, такі як ПЗ для резервного копіювання, можливості миттєвих знімків, функціонал дедуплікації даних, проміжне стиск, оптимізація роботи глобальної обчислювальної мережі (ГОМ) і багато чого іншого. Конвергентна інфраструктура в першу чергу опирається на апаратні засоби, а програмно-визначаємий ЦОД найчастіше адаптується під будь-яке апаратне забезпечення. У гіперконвергентній інфраструктурі ці дві можливості об'єднані.

Підтримка гіперконвергентної інфраструктури здійснюється одним постачальником. Нею можна управляти, як однією системою за допомогою єдиного набору інструментів. Для розширення інфраструктури просто потрібно встановити блоки необхідних пристроїв і ресурсів (наприклад, сховище) в основний системний блок. І робиться це буквально на лету.

Тому що гіперконвергентна інфраструктура є програмно-визначаємой (тобто, робота інфраструктури логічно відділена від фізичного встаткування),

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

віруси й інші види моніторингу у фоновому режимі (не відволікаючи користувачів).

Гнучкість гіперконвергентної інфраструктури робить її більше масштабованою й рентабельною, у порівнянні з конвергентною, оскільки в ній є присутнім можливість додавання обчислювальних пристроїв і пристроїв зберігання в міру необхідності. Вартість початкових капіталовкладень для обох інфраструктур висока, але в довгостроковій перспективі вартість вкладених коштів повинна окупитися.

На сьогоднішній день багато замовників у сфері середнього й великого бізнесу переходять до використання гіперконвергентних систем при розгортанні корпоративної IT-інфраструктури, і це обумовлено безумовними плюсами даних систем: легкістю масштабування і єдиним програмним інтерфейсом керування.

Гіперконвергентні системи поєднують у собі сервери, системи зберігання даних, мережний функціонал і програмне забезпечення. Такий підхід дозволяє знизити тимчасові й бюджетні витрати при створенні інфраструктури, а також спрощує експлуатацію системи.

По даним Gartner ринок гіперконвергентних платформ (HCI – Hyper-Converged Infrastructure) стрімко росте й досягне до 2026 року 5 мільярдів доларів. на даний момент HCI найбільш застосовні в сфері корпоративних ЦОД з кількох причин:

- Вирішують завдання масштабування IT-інфраструктури.
- Знижують витрати на розширення IT-інфраструктури.
- Зменшують кількість керованих систем.
- Збільшують швидкодію за рахунок використання SSD-Накопичувачів.
- Оптимізують програмну й апаратну платформи для спільної роботи.

Крім цього HCI дозволяють розгорнути програмно-конфігуруємі ЦОДи (SDDC).

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Архітектура й рішення різних виробників

НСІ, як правило, складаються з декількох модулів, що становлять горизонтально-масштабований кластер, що містить звичайно від 4 до 64 вузлів. Кожний з вузлів містить у собі обчислювальне ядро, систему зберігання, мережні ресурси й гіпервізор.

Крім цього, НСІ поставляються програмним забезпеченням для розподілу ресурсів системи й керування масштабуванням. ПЗ може бути встановлене заздалегідь або поставлятися окремо – більша частина НСІ-рішень поставляється із передінстальованими програмними компонентами.

Крім традиційних НСІ існують і рішення з референсною архітектурою, що дозволяє компонувати НСІ під конкретне завдання. Такі системи пропонують розроблювачі платформи EVO:RAIL. Система Glob від компанії Green IT може використовувати зовнішню СЗД, різні процесори, включаючи бюджетні варіанти Intel Xeon. Рішення Huawei FusionCube дозволяє вибирати блейд-вузли половинної або повної довжини з різними типами накопичувачів. Крім цього, деякі системи, наприклад, NetApp FlexPod використовують замість традиційних СЗД системи SDS.

Завдання й сценарії роботи

Більшість із існуючих НСІ здатні вирішувати широке коло завдань:

- Підтримка застосунків аналітики й систем OLTP.
- Підтримка критично важливих застосунків.
- Створення ресурсного пула на основі серверів і модернізація інфраструктури ЦОДа.
- Розгортання VDI і часток хмарних систем.

НСІ можна застосовувати для віртуалізованих застосунків, при цьому для різних віртуальних середовищ використовуються спеціальні конфігурації. Наприклад, для середовищ VMware оптимізовані системи VxRail, що використовують програмне забезпечення VMware vSphere і VMware VSAN. Цільовими завданнями цієї платформи є: аналітика «великих даних», Microsoft

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Exchange, САПР, наукові розробки, середовища вилучених філій і офісів, а також навантаження з підвищеними вимогами до графічних ресурсів.

Крім цього, виробники пропонують референсні системи для більше вузького класу навантажень, таких як додатка ERP і CRM.

Завдяки гнучкості рішень, великій кількості виробників і широкому вибору продуктів системи НСІ закономірно займають свою нішу в сфері великого бізнесу при розгортанні масштабних ЦОДо. Разом з тим, з'являються нові системи, які можуть бути актуальні для замовників у сфері середнього бізнесу, що робить ринок НСІ-рішень одним з тих, що найбільш розвиваються в ІТ-індустрії на сучасний момент.

3.2 Розробка структурної схеми

Недовга історія гіперконвергенції, що почалася із середини нульових років ХХІ століття, цікава своєю еволюцією. Початок був покладений утворенням альянсів декількома компаніями, що досить нетипово для ринку. Спільно вони запропонували так звані інтегровані стеки. Раніше інших – група VCE (VMware, Cisco й EMC) зі своїм Vblocks, потім група в складі VMware, Cisco і NetApp з мережею зберігання Fco і стеком FlexPod. Слідом був секстет у складі Dell, Fujitsu, HP, Microsoft, IBM і NEC з рішенням навколо гіпервізора Microsoft Hyper-V, а також дві самостійні компанії – Oracle з рішенням Exalogic і HP з Matrix.

Консолідовано виниклий рух в одному напрямку найбільших виробників змусило задуматися про фундаментальність передбачуваного зрушення.

І насправді, поява інтегрованих стеков змінило правила гри.[1] Раніше підприємству не залишалося нічого іншого, як збирати системи власними силами або прибїгати до послуг компаній-інтеграторів, а інтегрований стек можна було купити готовим, майже як домашній кінотеатр. Зрозуміло, тонкі цінителі й критики, яких у комп'ютерингу не мало, завжди знайдуть у будь-якому інтегрованому рішенні масу недоліків. Так було, наприклад, з інтегрованими пакетами ПЗ для ПК. Об'єднання редакторів, СУБД і електронних таблиць в

одному продукті приймалося ними в багнети доти, поки не з'явився один загальний Microsoft Office, на цьому всі дискусії закінчилися.

Суперечки ні, за окремими показниками готовий комплекс може уступати спеціалізованим рішенням, але в нього є головне достоїнство – можливість зсуву витрат з експлуатаційних (OpEx) у бік інвестиції в нові рішення (CapEx), тобто на підтримку виконання нового функціонала. Продовживши паралель із аудіо, можна сказати, що готові інтегровані стеки дозволяють витратити більше грошей на музику, а не на техніку для її відтворення.

Потім виникло нове узагальнене поняття «конвергентні інфраструктури» (Converged Infrastructure, CI).[2] Звичайно під CI розуміють цілісне закінчене рішення, зібране із взаємодоповнюючих готових компонентів: серверів, систем зберігання, комунікаційного встаткування й ПЗ для керування інфраструктурою, що включає засобу для автоматизації й оркестровки. У деяких випадках до складу CI можуть входити ще й спеціалізовані апаратні модулі керування. Керуюче ПЗ й спеціалізовані модулі забезпечують централізоване керування звичайно віртуалізованими й об'єднаними в загальні пули ресурсами, що розподіляються між різними додатками. Це підвищує економічну ефективність і знижує вартість експлуатації. Поява CI стало закономірним результатом еволюції корпоративних ЦОД у напрямку до приватних хмар.

За ступенем консолідованості CI можна розділити на три групи:

- шаблон, або референсна архітектура (Reference Architecture, RA);
- інфраструктура єдиного стека (Single Stack Infrastructure, SSI);
- істинно конвергентна інфраструктура (True Converged Infrastructure, TCI).

У першому випадку консолідація зводиться до окремих конфігурацій, які рекомендувалися, що складається з компонентів від різних виробників. У другому – весь стек устаткування виробляється й збирається в єдине ціле, а в третьому – у єдине ціле збираються компоненти від різних виробників, тобто в TCI об'єднані переваги RA і SSI.[3]

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

I, нарешті, ще один логічний крок. Його зробила компанія-стартап Nutanix, запропонувавши Complete Cluster – масштабовану інфраструктуру, що поєднує в собі складові частини корпоративної системи: комп'ютерні ресурси й ресурси систем зберігання у формі однакових базових модулів. Надалі саме такі інфраструктури стали називати гіперконвергентними (Hyperconverged Infrastructure, HCI).[4]

Відмінна риса Complete Cluster і всіх його послідовників складається в можливості збирати з модулів приватна корпоративна хмара без додаткових вкладень у мережні СЗД SAN або NAS. Ідея Complete Cluster, здавалося б, лежала на поверхні. Відомо, що успіх таких гігантів, як Google, Yahoo, Amazon і Facebook, значною мірою базується на власних блокових пропрієтарних технологіях, використовуваних для створення ЦОД. Чому б не застосувати аналогічні підходи до істотно меншим корпоративних ЦОД із класичними серверами й системами зберігання, адаптованими до традиційних навантажень? Такого роду запозичення досвіду має сенс, оскільки в сучасних корпоративних ЦОД, незалежно від їхнього масштабу, домінують близькі віртуалізовані навантаження.

Засновники Nutanix раніше інших звернули увагу на дві проблеми класичних ЦОДів, що складаються із серверів, SAN і NAS. Перша пов'язана зі спадщиною старих традицій – ЦОД створювалися розраховуючи на статичне навантаження, а нове подання про динамічне навантаження виникло у зв'язку з віртуалізацією серверів і переходом до хмар. Динаміка полягає в необхідності на лету створювати віртуальні машини, переміщати їх між серверами, однак у цих умовах керування існуючими мережними системами зберігання стає складн і незручним і відмахнутися від цієї проблеми вже не можна.

Росте кількість віртуальних машин, обсяг даних, якими вони оперують, збільшується, причому в чималому ступені кількісним змінам сприяють нові технології віртуалізації, зокрема, віртуалізація настільних систем.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Ще одна проблема сучасних ЦОДов виникає через те, що вони проектувалися з урахуванням невідповідності між усе більше зростаючою продуктивністю процесорів і повільністю, у силу їхньої механічної природи, жорстких дисків. Цей розрив приймався як даність, і його доводилося компенсувати за допомогою тих або інших методів розподілу даних по дисках. Однак з'явилися твердотільні накопичувачі (SSD) зі швидкістю обміну на два-три порядки вище, ніж у традиційних жорстких дисків (HDD), що могло б нівелювати розрив, але інфраструктура виявилася не підготовлена до них, і проста заміна HDD на SSD створює нову проблему – існуючі мережі не в змозі впоратися зі зростаючою потребою у швидкості передачі даних, що утрудняє використання переваг SSD.

В Nutanix відверто визнавали, що вони запозичили підхід до горизонтального масштабування, а їхній власний внесок складається в поширенні методики Google на корпоративні системи й пропозиції створення «готових до вживання» блоків для будівництва корпоративних систем. Для того щоб застосувати Complete Cluster до умов окремо взятого підприємства, загальну ідею довелося серйозно переробити. Якщо Google File System являє собою власне замовлене рішення, адаптоване для специфічних внутрішніх застосунків (пошук і електронна пошта), то Nutanix пропонує більше загальне рішення, адаптоване до умов корпоративної віртуалізованого середовища з упором на її особливі вимоги: ефективне керування даними, високу готовність, резервування, відновлення після збоїв.

Архітектура Nutanix Complete Cluster являла собою кластер з горизонтальним масштабуванням, зібраний з високопродуктивних вузлів. Окремий вузол має у своєму складі процесори, пам'ять і локальну систему зберігання, що складає з дисків HDD і SSD. На кожному вузлі працює власний екземпляр гіпервізора. Вузол служить хостом для працюючих на ньому віртуальних машин. Для об'єднання ресурсів окремих систем зберігання в єдиний віртуалізований пул служить SOCS (Scale-out Converged Storage) – аналог Google File System. Віртуальні машини за допомогою SOCS оперують даними точно так

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

само, як якби вони працювали з SAN, інакше кажучи, SOCS служить у якості гіпервізору систем зберігання, переміщаючи дані якнайближче до тої віртуальної машини, на якій вони використовуються, що підвищує продуктивність і знижує вартість. Крім того, Complete Cluster служить інструментом для горизонтального масштабування аж до декількох сотень вузлів.

Порівнюючи CI і HCI, можна використовувати будівельну аналогію. У першому випадку конструкцію збирають із визначеного набору великих блоків від зв'язаних між собою виробників, а в другому використовують підходящі з наявних на ринку цегл. Звичайно в створенні CI провідна роль належить тому зі членів альянсу, хто поставляє керування, але про інших, хто поставляє сервери, СЗД, мережі й віртуалізацію, теж не забувають. А системи класу HCI поставляються одним вендором, він збирає все необхідне в корпус (box) такими необхідними речами як засобу для резервного копіювання, створення снапшотів, дедуплікації, компресії в режимі in-line і т.буд. Конкретний состав набору в різних виробників може варіюватися, деякі, наприклад, SimpliVity, для прискорення використовують власні спеціалізовані інтегральні мікросхеми ASIC.

Ким би не була зроблена HCI, створюється віртуалізована інфраструктура, точніше кажучи, інфраструктура, віртуалізована по визначенню. Вона має наступні достоїнства:

- Гнучкість – реалізуються можливі типи масштабування.
- Економічна ефективність – знижуються як CAPEX, так і OPEX.
- Висока готовність – природне резервування блоків і швидка реконфігурація.
- Захищеність даних – дані не прив'язані до єдиного з погляду уразливості місцю.
- Ефективність роботи з даними – скорочуються витрати на СЗД і мережі.

В HCI реалізуються три типи масштабування: добре відоме по кластерах горизонтальне (scale out), по мейнфреймам і Unix-серверах вертикальне (scale up) і нове scale through.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Якщо порівнювати два підходи – гіпервіртуалізацію з дезагрегацією, то можна виразити обережне припущення про те, що в цих двох підходів можуть бути різні споживачі. У програм-гіпервізорів і в технології віртуалізації в цілому є межі, тому НСІ, швидше за все, не вийде за межі приватних хмар. У дезагрегації меж немає, тому її сфера застосування – ЦОДи компаній-гіперскейлерів, постачальників послуг глобальних хмар.

Програмні й мережні рішення для гіперконвергентних інфраструктур

Гіперконвергентні інфраструктури (НСІ) стають однією з найбільш перспективних технологій для створення приватних і гібридних хмар. Ідею НСІ запропонували стартапи, які раніше інших осознали можливість радикальних змін і що встигли зайняти лідируючі позиції в цьому сегменті ринку. Однак останнім часом і галузеві великовагові поїзди, що як звичайно затрималися на старті, але які володіють достатнім науковим і технічним потенціалом, роблять все можливе, щоб зайняти належне їм положення.

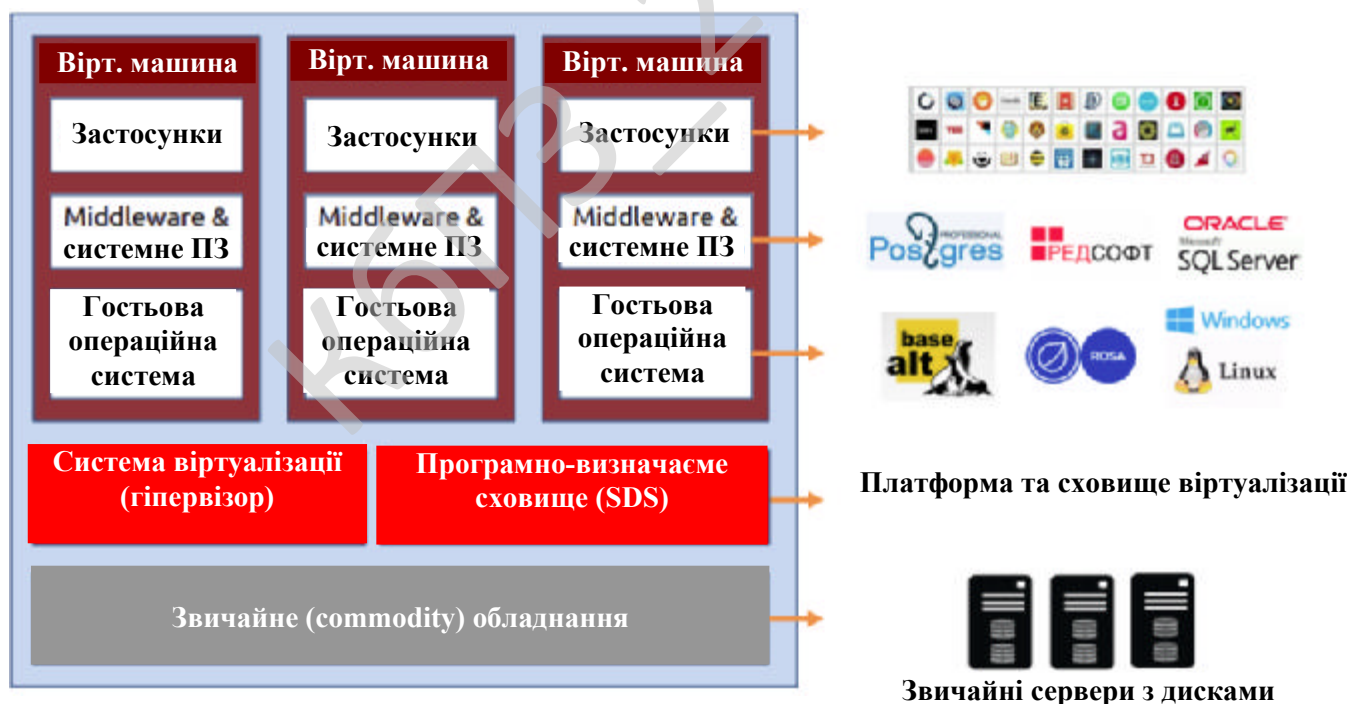


Рисунок 3.1 – Структурна схема системи

Загальна ситуація така: незалежно від того, під яким брендом виробляється та або інша конкурентна НСІ, вона складається із чотирьох взаємозалежних компонентів – сервери, СЗД, мережа й віртуалізація СЗД із метою створення загальних пулів (Storage Pool). Щодо цього, мабуть, єдине виключення являє собою зовсім що недавно з'явився трикомпонентний підхід до утворення НСІ на базі Intel Xeon Scalable із чип-сетом Intel C620 і СЗД Optane SSD.

Як постачальник НСІ, за рідкісним винятком, виступають компанії, що забезпечують функціонала, а саме два перших з перерахованих вище чотирьох компонентів. Вони можуть самі робити сервери й/або СЗД, або ж можуть здобувати їх, практично не маючи власного виробництва, тобто бути, як зараз говорять, фаблес.

3.3 Розробка функціональної схеми

Для організації системи гіперконвергентної інфраструктурної платформи використовується розподілені системи передачі даних (СПД). Розглянемо загальні принципи структурної й функціональної організації розподілених СПД системи гіперконвергентної інфраструктурної платформи, завдання проектування й різні методи дослідження СПД системи гіперконвергентної інфраструктурної платформи, а також проблеми розрахунку моделей СПД системи гіперконвергентної інфраструктурної платформи. Виконаємо огляд моделей і методів дослідження СПД системи гіперконвергентної інфраструктурної платформи. При цьому моделі СПД системи гіперконвергентної інфраструктурної платформи різних класів будуються на основі моделей масового обслуговування. Розглянуто класифікації систем (СМО) і мереж (СеМО) масового обслуговування. СМО використовуються як моделі каналів зв'язку, а СеМО – як моделі СПД системи гіперконвергентної інфраструктурної платформи. Для дослідження СПД системи гіперконвергентної інфраструктурної

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

платформи використовуються аналітичні (точні, наближені й евристичні) і імітаційні методи. Точні аналітичні методи використовуються для експонентних моделей, наближені методи використовуються для розрахунку неекспонентних моделей, розрахунок яких точними методами не можливий. Імітаційне моделювання реалізується в середовищі, використовуваному для дослідження складних систем з дискретним характером функціонування.

Розглянемо завдання приведення неоднорідного потоку до однорідного, розрахунку потоків пакетів у КЗ і оцінки пропускних здатностей каналів зв'язку СПД системи гіперконвергентної інфраструктурної платформи із урахуванням реальних особливостей.

Виявлено основні відмінності традиційного завдання оптимізації пропускних здатностей каналів зв'язку з однорідним навантаженням, припущеного Клейнроком, від завдання оптимізації з неоднорідним навантаженням, розглянутого в роботі.

Неоднорідність потоків пакетів описується параметрами різних типів повідомлень, до них відносяться:

- середня довжина повідомлень;
- середня зовнішня інтенсивність повідомлень від користувачів.

Крім того, при відомості неоднорідного потоку до однорідного необхідно враховувати:

- топологію СПД системи гіперконвергентної інфраструктурної платформи;
- метод маршрутизації;
- модель взаємодії користувачів;
- розподіл прикладних програм і наборів даних по вузлах мережі й т.д.

Розрахунок потоків пакетів у каналах СПД системи гіперконвергентної інфраструктурної платформи проводиться в три етапи.

На першому етапі виконується розрахунок інтенсивностей $\lambda_{i,j}$ пакетів з вузла i у вузол j шляхом множення зовнішніх інтенсивностей $\Lambda_{i,j}$ повідомлень із вузла i у вузол j на число пакетів $k = 1, 2, \dots$ в одному повідомленні:

$$\lambda_{i,j} = k\Lambda_{i,j} \quad (i, j = \overline{1, n}; \quad i \neq j), \quad (3.1)$$

На другому етапі розрахунок інтенсивностей пакетів у КЗ виконується на основі обчислених на першому етапі зовнішніх інтенсивностей потоків пакетів, заданих таблиць маршрутизації й імовірності передачі по основному шляху:

$$\lambda_{(k,l)} = \sum_{i=1}^n \sum_{j=1}^n \lambda_{i,j} X_{(k,l)}^{(i,j)} \quad (k, l = \overline{1, n}; \quad k \neq l), \quad (3.2)$$

де $X_{(k,l)}^{(i,j)}$ – частка пакетів ($0 \leq X_{(k,l)}^{(i,j)} \leq 1$) що проходять по каналу (k, l) , причому повинне виконуватися умова збереження потоку в мережі:

$$\sum_{k=1}^n X_{(k,l)}^{(i,j)} - \sum_{k=1}^n X_{(l,k)}^{(i,j)} = \begin{cases} -1, & l=i; \\ 0, & l \neq i; \\ 1, & l=j. \end{cases} \quad (3.3)$$

Для визначення $X_{(k,l)}^{(i,j)}$ використовується зворотна рекурсивна процедура, визначаюча розподіл інтенсивностей потоків на парі (i – вузол-джерело, j – вузол-призначення) і підсумовуюча їх при розгляді всіх пар вузлів. Ця процедура реалізується в такий спосіб. Скануються всі вузли, через які пакети можуть пройти від вузла-джерела до вузла-призначення з урахуванням таблиць маршрутизації, і будується двійкове дерево, причому для кожного вузла розглядаються два шляхи передачі пакета: основний і альтернативний. СПД системи гіперконвергентної інфраструктурної платформи відрізняється від двійкового дерева тим, що її топологія довільна. Для того, щоб усунути ситуацію, коли пакети можуть циркулювати в мережі нескінченно, використовується маскування вузлів, які вже сканувалися.

При виклику рекурсивної процедури частка пакетів у кожному з каналів зв'язку розраховується за правилом:

1. Якщо наступний розглянутий вузол є основним, і він не маскований, то частка пакетів у цьому каналі збільшується на величину, рівну добутку ймовірності передачі пакетів по основному шляху на частку пакетів попереднього КЗ.

2. Якщо наступний розглянутий вузол є альтернативним і він не маскований, то частка пакетів у цьому каналі збільшується на величину, рівну добутку ймовірності передачі по альтернативному шляху на частку пакетів попереднього каналу зв'язку.

Рекурсивна процедура викликається тільки тоді, коли наступний розглянутий вузол у процесі сканування не маскований і не збігається з вузлом-призначення. Зворотний механізм у рекурсивному алгоритмі реалізується за допомогою масиву маскування – після виклику рекурсивної процедури якоїсь пари (вузол-джерело, вузол-призначення) потрібно відзначити вузол-джерело як немаскований.

Якщо основний шлях розглянутого вузла збігається з вузлом-призначення й альтернативний шлях цього вузла є маскованим вузлом, то до частки пакетів основного каналу зв'язку додається частка пакетів альтернативного, рівна добутку ймовірності передачі пакета по альтернативному шляху на частку пакетів попереднього каналу зв'язку, тому що пакети не можуть повторно попадати в той самий вузол СПД системи гіперконвергентної інфраструктурної платформи.

На третьому етапі виконується розрахунок коефіцієнтів передач для кожного каналу зв'язку:

$$\alpha_{(k,l)} = \lambda_{(k,l)} / \lambda_{\text{вн.}}, \quad (3.4)$$

де $\lambda_{\text{вн.}} = \sum_{i=1}^n \sum_{j=1}^n \lambda_{i,j}$ – сумарна інтенсивність надходження пакетів у СПД системи гіперконвергентної інфраструктурної платформи.

Після розрахунку інтенсивностей пакетів у КЗ для кожного типу повідомлень із використанням вищенаведеного алгоритму неоднорідний потік

пакетів зводиться до однорідного шляхом підсумовування інтенсивностей потоків пакетів у КЗ.

Даний алгоритм розрахунку інтенсивностей пакетів вбудований у програмний комплекс і являє собою ключовий етап для рішення завдання оцінки пропускних здатностей на основі методу Лагранжа.

Представлено завдання оптимізації пропускних здатностей каналів зв'язку на основі методу Лагранжа при обмеженні на середній час доставки пакетів у СПД системи гіперконвергентної інфраструктурної платформи і на вартість СПД системи гіперконвергентної інфраструктурної платформи. Наведено алгоритм вибору дискретних значень пропускних здатностей на основі отриманих безперервних значень.

Досліджуємо вплив процесів передачі даних на характеристики функціонування СПД системи гіперконвергентної інфраструктурної платформи.

Виконано численні імітаційні експерименти для оцінки погрішностей аналітичних методів розрахунку моделі каналу зв'язку, що базуються на наближених формулах розрахунку часових характеристик каналу зв'язку.

Експерименти дозволяють зробити вивід про те, що наближена формула розрахунку часових характеристик системи G/M/1, використовуваної як модель КЗ, дає невелику погрішність для потоку, коефіцієнт варіації (КВ) якого менше 1 (у більшості випадків вона менше 20% при завантаженні більше 0.3). Коефіцієнт варіації – це відношення середньоквадратичного відхилення до середньої арифметичної, який показує ступінь відхилення отриманих значень. Для потоку, КВ якого більше 1, при збільшенні завантаження зменшення цієї погрішності близько до лінійного. При великому завантаженні каналу зв'язку погрішність розрахунку характеристик передачі пакетів перебуває в межах 20%, прийнятних для інженерних розрахунків. Численні імітаційні експерименти дозволили виконати детальне дослідження характеристик системи й зробити наступні висновки:

1. Чим менше завантаження досліджуваної системи, тим більшу погрішність дає наближена формула.

2. При зміні завантаження системи за рахунок варіювання середнього часу обслуговування (при фіксованому середньому часі надходження) і за рахунок варіювання середнього часу надходження (при фіксованому середньому часі обслуговування) експерименти дають однаковий результат, тобто результати розрахунку погрішностей характеристик не залежать від того, за рахунок якого параметра змінюється завантаження.

Формула розрахунку коефіцієнтів варіацій вихідного потоку каналу зв'язку дає в більшості випадків верхню границю для коефіцієнта варіації вихідного потоку заявок із системи G/G/1, причому відносна погрішність формули перебуває в прийнятних межах (менше 20%) при малому завантаженні системи (від 0 до 0.4) для коефіцієнтів варіації вхідного потоку заявок більше 1, а також при великому завантаженні системи (від 0.8 до 1) для випадків KV менше 1.

Розглянуто різні підходи по оцінці впливу обмеженої довжини пакетів на характеристики каналів окремо й у мережі в цілому. Показано, що для нормування усіченого експонентного розподілу обидва методи (імітаційний і комбінація аналітичного і імітаційного) дають однакові результати. Для експонентного розподілу обидва способи обмеження пакетів ліворуч і праворуч (усіканням і зсувом) дають практично однакові часові характеристики передачі даних. Однак, для гіперекспонентного розподілу спосіб «зі зсувом» дає значення часу очікування значно більші, ніж для випадку «з усіканням».

Для мережі Ethernet відносне відхилення результатів розподілу зі зсувом у порівнянні із традиційними становить десятки відсотків, і це відхилення збільшується зі збільшенням завантаження каналу й може перевищувати 100% для гіперекспоненційного розподілу. Традиційні розподіли можуть використовуватися при моделюванні мережі TCP/IP через велике відношення між максимальною l_{\max} (мінімальною – l_{\min}) і середньою l_{cp} довжиною пакетів.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

Розходження в результатах для СПД системи гіперконвергентної інфраструктурної платформи, у загальному випадку, може становити десятки відсотків, що робить необхідним облік обмеженої довжини пакетів у процесі системотехнічного проектування СПД системи гіперконвергентної інфраструктурної платформи.

Проведено аналіз впливу характеру трафіку в СПД системи гіперконвергентної інфраструктурної платформи і тривалості передачі пакетів у каналах зв'язку на характеристики функціонування СПД системи гіперконвергентної інфраструктурної платформи. Коли всі КВ розподілу потоків і часу передачі пакетів у каналах збільшуються до одиниці, всі часові характеристики неекспонентної мережі прагнуть до часових характеристик експонентної мережі, тобто можна говорити, що експонентна мережа розглядається як верхня межа в цьому випадку. Коли значення КВ різних розподілів збігаються, характеристики мережі майже збігаються. У мережах з гіперекспонентним розподілом часу обслуговування (передачі) пакетів у каналах СПД системи гіперконвергентної інфраструктурної платформи забезпечується менше значення часу доставки пакетів у порівнянні з Гамма-розподілом, причому при більших значеннях КВ це розходження може становити десятки відсотків. При більших значеннях КВ інтервалів часу між вступниками в мережу пакетами й часу їхньої передачі по каналах зв'язку необхідно враховувати більше високі моменти розподілів, зокрема третій момент, що обумовлює значне розходження часу доставки пакетів у СПД системи гіперконвергентної інфраструктурної платформи.

Розглянуто вплив третього моменту гіперекспонентного (із КВ більше 1) і гіпоекспонентного (із КВ менше 1) розподілу часу обслуговування (передачі пакетів у каналі) на характеристики системи. Для гіперекспонентного закону з одними й тим же другим моментом можна одержати різні значення третього моменту, і чим більше значення другого моменту, тим більше значення третього моменту і їхнє розходження. Залежності часу очікування систем $H_2/M/1$, $H_2/D/1$, $H_2/\Gamma_2/1$, (де Γ_2 – гамма-розподіл з коефіцієнтом варіації 2) від величин третього

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

моменту гіперекспонентного потоку для завантаження 0.5 і 0.9, показані. Чим більше значення третього моменту, тим менше час очікування в системі. Однак, коли завантаження дорівнює 0.5, час очікування різних систем різко падає долілиць при незначній зміні третього моменту на початку (коли значення третього моменту <100) і майже не змінюється, коли ці значення більше 100. Для випадку, коли завантаження дорівнює 0.9, час очікування систем більше «лінійно» залежить від третього моменту. Для всіх випадків, час очікування сильно залежить від третього моменту й дає велике розходження (більше 100% при завантаженні 0.5).

Дослідження впливу третього моменту поступаючих даних в систему $E_x/G/1$ потоку на час очікування дає результати, показані впливом третього моменту гіпоекспоненційного вступника потоку з різними КВ на відносне розходження між максимальним і мінімальним значеннями часу очікування декількох систем при $k = 50$.

Функціональна схема розробленої системи зображена на рисунку 3.2. Представимо опис розробленого програмного комплексу для дослідження СПД системи гіперконвергентної інфраструктурної платформи із неоднорідним потоком повідомлень на основі аналітичного й імітаційного моделювання.

Завдання проектування мережі складається у визначенні пропускних здатностей КЗ із урахуванням неоднорідності трафіку, різноманіття топологій, алгоритмів маршрутизації, варіантів розміщення прикладних програм і наборів даних по вузлах мережі, способів взаємодії користувачів мережі.

Аналітична модель СПД системи гіперконвергентної інфраструктурної платформи будується у вигляді розімкнутої СеМО для будь-якої топології мережі, що задається аналітично або графічно. На основі аналітичної моделі вирішується завдання визначення пропускних здатностей КЗ у розподілених СПД системи гіперконвергентної інфраструктурної платформи при обмеженнях на час доставки пакетів або на вартість мережі.

Програма дозволяє розрахувати характеристики СПД системи гіперконвергентної інфраструктурної платформи, такі як:

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- пропускні здатності КЗ;
- час затримки пакетів при передачі по кожному каналу й у мережі в цілому;
- завантаження кожного КЗ;
- інтенсивності потоків пакетів у каналах зв'язку;
- імовірності передачі пакетів від користувачів у мережу, між каналами й з мережі до користувачів.

Програма дозволяє варіювати отримані характеристики, такі як:

- пропускні здатності;
- час передачі по КЗ;
- завантаження КЗ;
- час доставки пакетів у СПД системи гіперконвергентної інфраструктурної платформи;
- вартість СПД системи гіперконвергентної інфраструктурної платформи для одержання необхідних результатів.

Крім того, вона дозволяє вибрати значення пропускних здатностей з дискретного ряду значень, знайдених із традиційної оптимізації безперервних значень пропускних здатностей.

Імітаційна модель СПД системи гіперконвергентної інфраструктурної платформи генерується автоматично на основі розрахунку характеристик аналітичної моделі у вигляді розімкнутої експонентної СеМО. Отримані характеристики аналітичної моделі використовуються як параметри імітаційної моделі СПД системи гіперконвергентної інфраструктурної платформи, що представляє собою розімкнуту СеМО, вузли якої відповідають каналам зв'язку, а заявки – пакетам у СПД системи гіперконвергентної інфраструктурної платформи. Число джерел заявок в імітаційній моделі дорівнює числу вузлів у СПД системи гіперконвергентної інфраструктурної платформи, при цьому інтенсивності надходження заявок у моделі визначаються як зовнішні інтенсивності пакетів від користувачів до вузлів СПД системи гіперконвергентної інфраструктурної платформи відповідно.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Аналітична модель, реалізована в програмі, дозволяє розрахувати наступні параметри для імітаційної моделі.

1. Імовірність передачі пакетів від користувача j до каналу k .
2. Імовірність передачі пакетів від каналу k до каналу h .
3. Імовірність передачі пакетів від каналу k до користувача j .
4. Середній інтервал часу між вступниками пакетами від користувача j у СПД системи гіперконвергентної інфраструктурної платформи.
5. Середній час передач пакетів у каналах зв'язку k .

Крім розрахованих параметрів для імітаційної моделі СПД системи гіперконвергентної інфраструктурної платформи додатково необхідно задати наступні параметри:

а) закони розподілів інтервалів часу між вступниками пакетами від користувачів у СПД системи гіперконвергентної інфраструктурної платформи із коефіцієнтами варіації цих розподілів;

б) закони розподілів часу передачі пакетів у КЗ із коефіцієнтами варіації цих розподілів.

Відзначимо, що імітаційна модель СПД системи гіперконвергентної інфраструктурної платформи призначена для детального аналізу характеристик функціонування мережі, спроектованої в процесі аналітичного моделювання. При цьому, у випадку відмінності реального характеру процесів надходження пакетів у мережу або передачі пакетів по каналах зв'язку від експонентного, в імітаційній моделі передбачена можливість варіювання законів розподілу часу передачі (довжин обслуговування) пакетів у кожному із КЗ, а також законів розподілу інтервалів часу між вступними в мережу пакетами. Як такі закони в роботі використовувалися наступні розподіли: експонентний, детермінований; гіпоекспоненційний різного порядку й, відповідно, з різними коефіцієнтами варіації; рівномірний; експонентний з ненульовими зсувами; гіперекспонентний; Гамма-розподіл.

Аналогічно розроблений засіб дослідження часових характеристик каналу зв'язку шляхом генерування імітаційної моделі каналу зв'язку у вигляді СМО

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

типу G/G/1 з можливістю зміни завантаження каналу й варіювання законів розподілу інтервалів часу між пакетами й часу передачі пакетів по каналі. Даний засіб дозволяє одержати значення часових характеристик каналу зв'язку й оцінити погрішність аналітичних методів розрахунку характеристик каналу зв'язку.

Сформульовано методика дослідження розподіленої СПД системи гіперконвергентної інфраструктурної платформи із неоднорідним трафіком, що містить наступні етапи.

1. Підготовка вихідних даних для програмного комплексу:

- кількість вузлів мережі і їхнє взаємне розташування;
- навантаження СПД системи гіперконвергентної інфраструктурної платформи, створювана користувачами при роботі із прикладними програмами, наборами даних і в процесі обміну повідомлень;
- обмеження на середній час доставки пакетів або на вартість СПД системи гіперконвергентної інфраструктурної платформи;
- максимальна довжина пакета;
- вибір типу каналів і завдання вартісних коефіцієнтів КЗ.

2. Розрахунок оптимальних пропускних здатностей типових топологій СПД системи гіперконвергентної інфраструктурної платформи при заданих наступних параметрах:

- типова топологія: зірка, кільце, дерево, повнозв'язна;
- модель взаємодії користувачів мережі: RDA (Remote Data Access), DBS (DataBase Server) і AS (Application Server);
- розподіл прикладних програм і наборів даних по вузлах СПД системи гіперконвергентної інфраструктурної платформи;
- метод маршрутизації.

Після завдання цих параметрів розраховуються оптимальні значення пропускних здатностей КЗ, час передачі пакетів і завантаження каналів.

Аналітична модель СПД системи гіперконвергентної інфраструктурної платформи

Блок розрахунку характеристик СПД системи гіперконвергентної інфраструктурної платформи

- пропускна здатність КЗ;
- час затримки пакетів при передачі по кожному каналу й у мережі в цілому;
- завантаження кожного КЗ;
- інтенсивності потоків пакетів у каналах зв'язку;
- імовірності передачі пакетів від користувачів у мережу, між каналами й з мережі до користувачів.

Блок варіювання характеристик СПД системи гіперконвергентної інфраструктурної платформи

- пропускна здатність КЗ;
- час передачі по КЗ;
- завантаження КЗ;
- час доставки пакетів у СПД;
- вартість СПД системи гіперконвергентної інфраструктурної платформи для одержання необхідних результатів.



Блок дослідження СПД системи гіперконвергентної інфраструктурної платформи

Підготовка вихідних даних для програмного комплексу

Детальний аналіз спроектованої СПД

Розрахунок оптимальних пропускних здатностей типових топологій СПД при заданих параметрах

Вибір найкращої топології СПД, для якої критерій ефективності СПД приймає найменше значення.

Розрахунок пропускних здатностей КЗ при розподілених топологіях

Оцінка впливу характеру потоків пакетів і тривалості передачі даних на СПД

Варіювання параметрів і характеристик КЗ і СПД і вибір дискретних значень пропускних здатностей

Дослідження СПД при різних розміщеннях прикладних програм і наборів даних

Оцінка погрешностей аналітичних методів розрахунку

Імітаційна модель СПД системи гіперконвергентної інфраструктурної платформи

Блок розрахованих параметрів СПД

Блок завдання параметрів СПД

Блок дослідження часових характеристик КЗ



Рисунок 3.2 – Функціональна схема системи

3. Розрахунок пропускних здатностей КЗ при розподілених топологіях.

4. Вибір найкращої топології СПД системи гіперконвергентної інфраструктурної платформи, для якої обрана залежно від постановки завдання як критерій ефективності характеристика СПД системи гіперконвергентної інфраструктурної платформи (середній час доставки пакетів або вартість СПД системи гіперконвергентної інфраструктурної платформи) приймає найменше значення.

5. Варіювання параметрів і характеристик КЗ і СПД системи гіперконвергентної інфраструктурної платформи і вибір дискретних значень пропускних здатностей.

6. Оцінка погрішностей аналітичних методів розрахунку характеристик КЗ.

7. Оцінка впливу характеру потоків пакетів і тривалості передачі даних на характеристики СПД системи гіперконвергентної інфраструктурної платформи:

– вплив третього моменту розподілу вхідного потоку пакетів на характеристики КЗ;

– вплив довжини пакетів на характеристики каналів зв'язку й СПД системи гіперконвергентної інфраструктурної платформи;

– вплив законів розподілів трафіку в мережах і часі передачі пакетів у КЗ на характеристики функціонування СПД системи гіперконвергентної інфраструктурної платформи.

8. Дослідження СПД системи гіперконвергентної інфраструктурної платформи при різних розміщеннях прикладних програм і наборів даних.

9. Детальний аналіз спроектованої СПД системи гіперконвергентної інфраструктурної платформи, що припускає варіювання параметрів СПД системи гіперконвергентної інфраструктурної платформи, у тому числі: довжини запитів і відповідей прикладних програм і наборів даних, імовірність передачі по основному шляху, способи маршрутизації й т.д.

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

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.



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.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію магістерської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень. Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

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

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

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

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

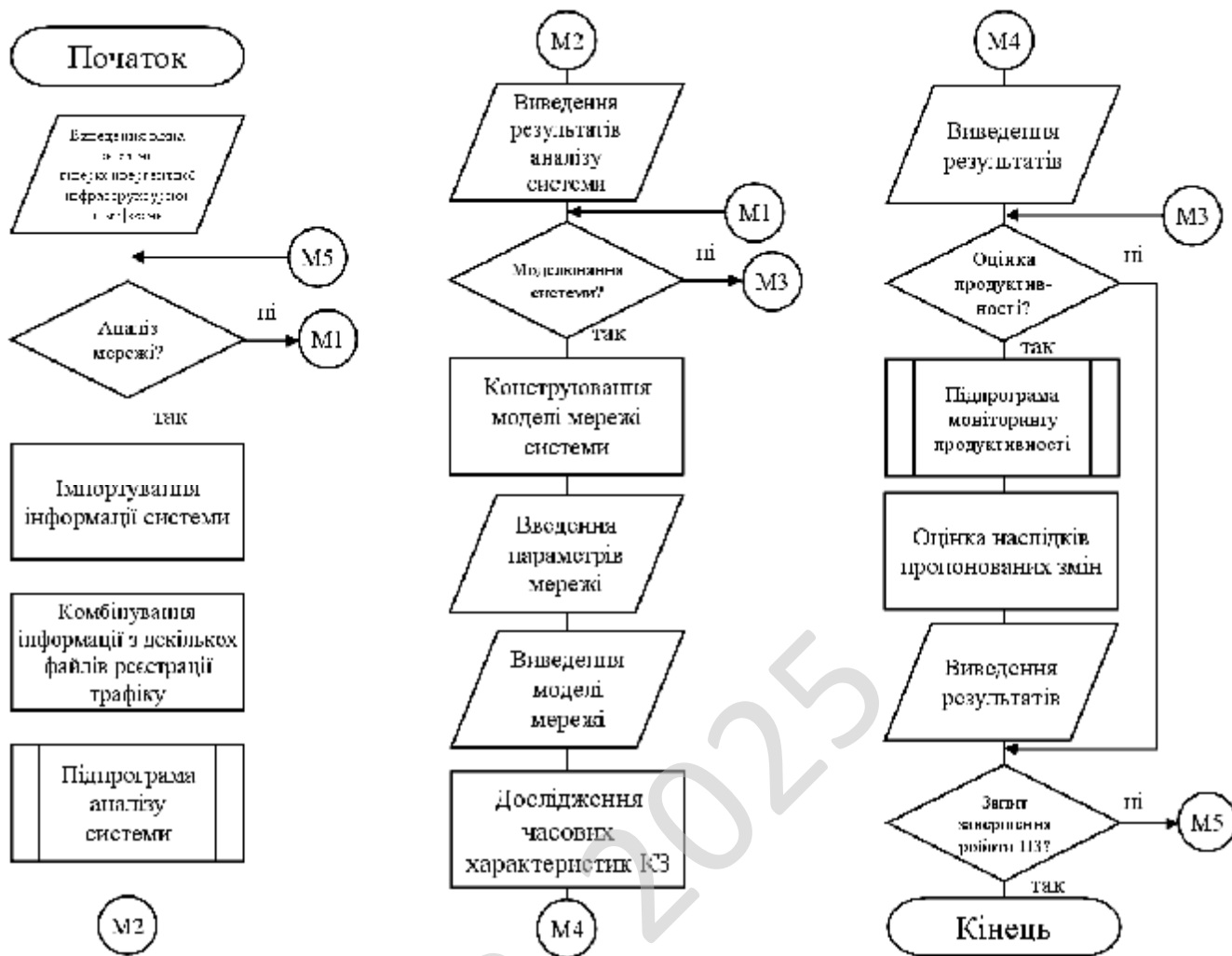


Рисунок 4.1 – Блок-схема основної програми

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, названої 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.

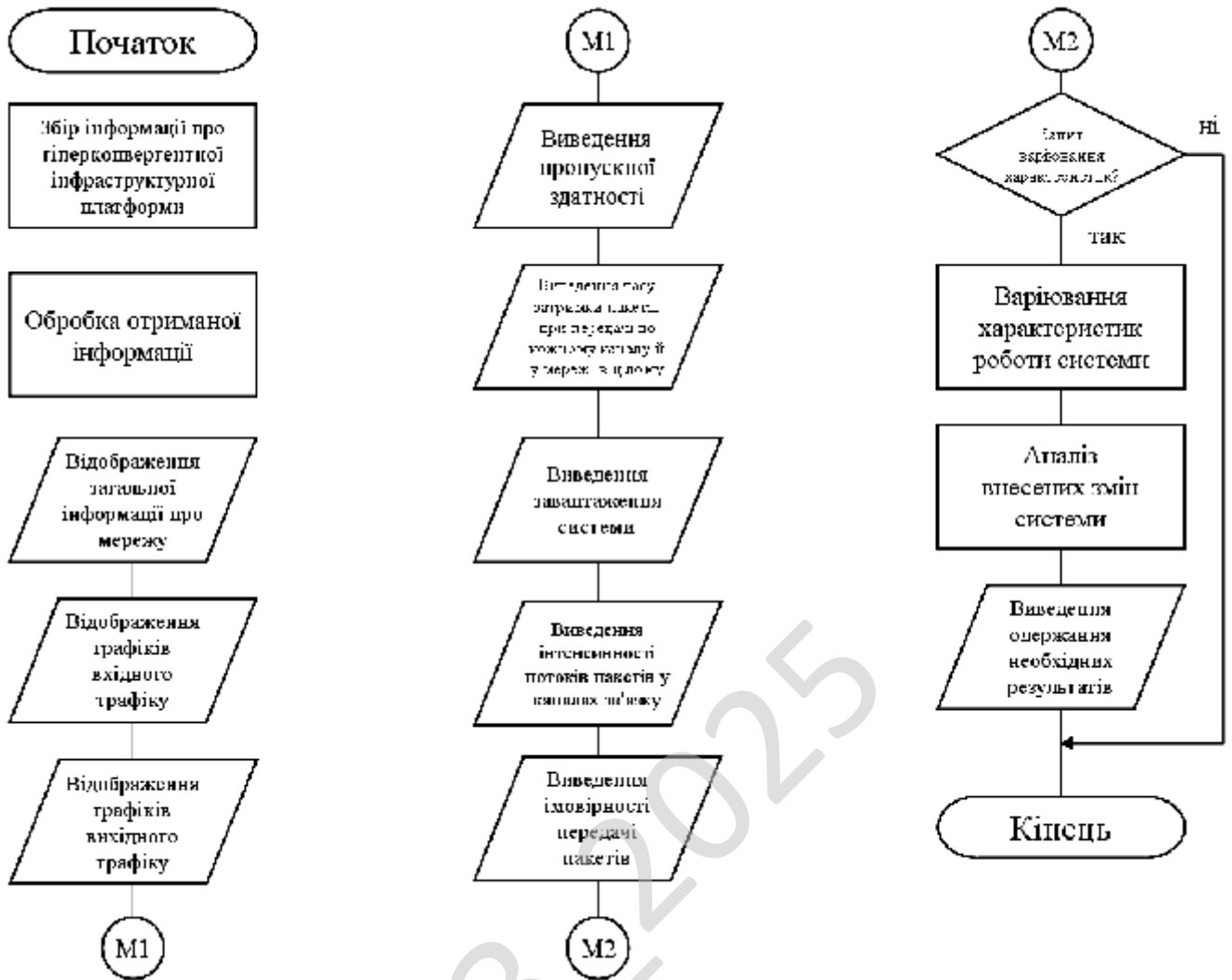


Рисунок 4.2 – Блок-схема роботи підпрограми

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

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

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до

серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

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

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

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

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного.

В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

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

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

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

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

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

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

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

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

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

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

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

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

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

Розширювана мова розмітки (Extensible Markup Language, скорочено XML) – запропонований консорціумом World Wide Web Consortium (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет. Є спрощеною підмножиною мови розмітки SGML.

XML-документ складається із текстових знаків, і придатний до читання людиною. Стандарт XML (Recommendation, перше видання від 10 лютого 1998, останнє, четверте видання 29 вересня 2006) визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів.

Цей формат достатньо гнучкий для того, аби бути придатним для застосування в різних галузях. Іншими словами, запропонований стандарт визначає метамову, на основі якої шляхом запровадження обмежень на структуру та зміст документів визначаються специфічні, предметно-орієнтовані мови розмітки даних. Ці обмеження описуються мовами схем (Schema), такими як XML Schema (XSD), DTD або RELAX NG.

Прикладами мов, заснованих на XML, є: XSLT, XAML, XUL, RSS, MathML, GraphML, XHTML, SVG, а також XML Schema.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Основні поняття. Коректність.

Коректний документ (well-formed document) відповідає всім синтаксичним правилам XML. Документ, що не є коректним, не може називатись XML-документом. Сумісний синтаксичний аналізатор (Conforming parser) не повинен обробляти такі документи. Зокрема, коректний XML-документ має:

1. Лише один елемент у корені.
2. Непорожні елементи розмічено початковим та кінцевим тегами (наприклад, <пункт> Пункт 1</пункт>). Порожні елементи можуть позначатися «закритим» тегом, наприклад <IAmEmpty />. Така пара еквівалентна <IAmEmpty></IAmEmpty>.
3. Один елемент не може мати декілька атрибутів з однаковою назвою. Значення атрибутів перебувають або в одинарних (') , або у подвійних (") лапках.
4. Теги можуть бути вкладені, але не можуть перекриватись. Кожен некореневий елемент мусить повністю перебувати в іншому елементі.
5. Документ має складатися тільки з правильно закодованих дозволених символів Юнікоду. Єдиними кодуваннями, які обов'язково має розуміти XML-процесор, є UTF-16 та UTF-8. Фактичне та задеклароване кодування (character encoding) документа мають збігатись. Кодування може бути задекларовано ззовні, як у заголовку «Content-Type» при передачі по протоколу HTTP, або в самому документі використанням явної розмітки на самому початку документа. У разі відсутності інформації про кодування, документ має бути в кодуванні UTF-8 (або його підмножині ASCII).

Валідність

Документ називається валідним (valid), якщо він є коректним, містить посилання на граматичні правила та повністю відповідає обмеженням, вказаним у цих правилах (DTD або XML Schema або іншому подібному документі).

Синтаксичний аналізатор

Синтаксичним аналізатором (часто парсер, від parser) називається програма або компонент, що читає XML-документ, проводить синтаксичний

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

аналіз та відтворює його структуру. Якщо синтаксичний аналізатор перевіряє документ на валідність, то такий аналізатор називають валідатором (validating).

Назви елементів чутливі до регістру літер. Наприклад, наступна пара елементів правильна: `<Step> ... </Step>` у той час як ця – ні: `<Step> ... </step>`.

Правильний вибір назв для XML-елементів підкреслюватиме значення даних у створеній мові розмітки. Це сприятиме полегшенню роботи людей з такими документами, зберігаючи можливості для комп'ютерної обробки даних.

Вибір змістовних назв передає семантику елементів та атрибутів для людини, без посилання на зовнішню документацію. Однак це може призвести до надмірності розмітки, що ускладнює редагування й збільшує розмір файлів.

Правильний вибір назв для XML-елементів підкреслюватиме значення даних у створеній мові розмітки. Це сприятиме полегшенню роботи людей з такими документами, зберігаючи можливості для комп'ютерної обробки даних. Вибір змістовних назв передає семантику елементів та атрибутів для людини, без посилання на зовнішню документацію. Однак це може призвести до надмірності розмітки, що ускладнює редагування й збільшує розмір файлів. XML-документи мають як фізичну, так і логічну структуру.

Приклад XML-документа:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.3/" xml:lang="uk">
  <page>
    <title>Фукідід</title>
    <id>1529</id>
    <revision>
      <id>4382</id>
      <timestamp>2026-09-18T22:11:53Z</timestamp>
      <minor />
      <comment>Interwiki</comment>
      <text xml:space="preserve">{{Wikipedia}}
    </text>
    </revision>
  </page>
</mediawiki>
```

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Фізична структура

Сутності (Entity). Головною сутністю є зміст документа. Інші можливі сутності вказуються за допомогою. Посилання на сутності (&назва; в самому документі, та, наприклад %назва; у визначенні його типу) можуть слугувати в ролі позначень спеціальних символів, посилань на спеціальні символи або окремих документів чи фрагментів тексту.

XML-декларація, в ній вказується версія XML, кодування та інша допоміжна інформація.

Декларація типу документа може застосовуватись для того, аби додавати нові типи сутностей та визначати логічну структуру документа.

Логічна структура

XML-документ має ієрархічну логічну структуру, і може представлятись у вигляді дерева. Вузлами цього дерева можуть бути: елементи, фізична структура яких складається із коректної пари відкриваючого та закриваючого тегів (<Назва-тега>) та (</Назва-тега>), або тега порожнього елемента (<Назва-тега />).

Атрибути, що мають вигляд пар ключ-значення (назва атрибута="значення атрибута") і знаходяться або у відкриваючому, або у порожньому тезі (подібно до метаданих).

Вказівки щодо обробки документа (Processing Instruction) (<?Обробник параметр ?>).

Коментарі (<!-- Текст коментаря -->).

Текст, або у вигляді простого тексту, або фрагментів CDATA (<![CDATA[довільний текст]]>).

XML-документ повинен мати лише один кореневий елемент. Решта елементів є піделементами цього кореневого елемента. Деякі веб-браузери здатні безпосередньо відобразити XML-документи. Це може досягатись шляхом застосування таблиці стилів (Stylesheet). Вказані у таблиці стилів операції можуть призводити до перетворення XML-документа в інший, відмінний від XML формат.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Коректність XML-документів

Залишивши назви, дозволена ієрархію, та значення елементів і атрибутів відкритою та можливою бути визначеною в спеціалізованих схемах або визначеннях типу документа (DTD).

XML утворює синтаксичну основу для створення спеціалізованих, заснованих на XML мовах розмітки даних.

Загальний синтаксис таких документів стабільний і наперед визначений – документи мають відповідати базовим вимогам XML, гарантуючи те, що довільне програмне забезпечення з підтримкою XML буде здатне щонайменше зчитати і відтворити відносну структуру інформації, що міститься в них.

Схема лише доповнює синтаксичні правила множиною обмежень. Зазвичай схеми обмежують назви елементів та атрибутів, дозволені типи значень і допустиму ієрархію елементів, наприклад, дозволяючи лише елементу з назвою «народження» містити під-елемент з назвою «місяць» та з назвою «день», і кожен із них мусить містити лише літери. Обмеження, вказані в схемі, можуть також включати присвоєння певних типів даних для впливу на те, як обробляється інформація; наприклад, дані елемента «місяць» можна визначити як такі, що містять лише місяць, як це визначено відповідно до використаної мови схем.

Коректний XML-документ, що відповідає обмеженням схеми або DTD, називається валідним.

DTD (Document Type Definition). Найдавнішим форматом схем для XML є успадкований від SGML формат визначення типу документа (Document Type Definition, DTD). У той час, як через включення до стандарту XML 1.0 DTD став поширеним форматом схем, він має такі обмеження:

1. Відсутність нових можливостей XML, із найважливішою з посеред них простори назв.
2. Брак виразності. Деякі формальні аспекти XML-документів неможливо відобразити в DTD.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

3. Використовується спеціалізований, заснований не на XML синтаксис для опису схем.

DTD все ще використовується в багатьох програмах, оскільки він вважається найпростішим форматом для аналізу та збереження.

XML Schema. На заміну DTD було розроблено нову мову схем – XML Schema (буквально – XML-схема), скорочено позначається як XSD (від XML Schema Definition). XSD набагато потужніші за DTD в описанні заснованих на XML мов. Вони використовують багатий набір типів даних, підтримують детальніші обмеження на структуру документів, та повинні оброблятися складнішими системами. XSD побудовано на основі XML, що робить можливим використання звичайних інструментів XML для їхньої обробки, хоча реалізації XSD вимагають набагато більше ніж просто можливість читати XML.

Серед недоліків XSD називають такі:

1. Специфікація дуже велика, що робить її складною для розуміння та реалізації.

2. Оснований на XML синтаксис додає надмірності мові, що ускладнює читання та запис XSD.

3. Валідація відносно схеми може бути дорогим додатком до синтаксичного аналізу XML-документів.

4. Можливості моделювання дуже обмежені, без можливості впливу значень атрибутів на вміст елементів.

5. Модель отримання типів даних є дуже обмеженою, зокрема в тому, що отримання шляхом розширення є рідко коли корисним.

6. Механізми ключа/посилання на ключ/унікальності не враховують тип даних.

7. Концепція PSVI (Post Schema Validation Infoset) не має стандартного представлення або прикладного програмного інтерфейса, що працює проти незалежності від реалізації, якщо не виконується повторна валідація.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

RELAX NG є іншою поширеною мовою схем для XML. Вперше RELAX NG було визначено стандартом OASIS, а тепер – міжнародним стандартом ISO (як частина DSDL). Ця мова схем має два формати: оснований на XML, та компактний, не-XML. Компактний синтаксис призначений для покращення можливості читання та написання схем, однак, оскільки існує точно визначений спосіб перетворення компактного формату в оснований на XML, і навпаки, не втрачаються переваги від використання стандартних XML-інструментів. RELAX NG має простіші системи для визначення та валідації у порівнянні з XML Schema, що робить її привабливішою для використання та реалізації. Також існує можливість використання модулів роботи з типом даних; наприклад, автор схеми RELAX NG може вказати, що значення XML-документа мають відповідати визначенням типам даних у форматі XML Schema Datatypes.

ISO DSDL та інші мови схем

Стандарт ISO DSDL (Document Schema Description Languages, мови описання схем документів) об'єднує широке коло малих мов схем, кожна із яких призначена для розв'язання окремих проблем.

До DSDL належить RELAX NG із повним та компактним синтаксисом, мова припущень Schematron, та мови для визначення типів даних, обмежень на літери, перейменування та розкривання мнемонік, та основане на просторах назв перенаправлення фрагментів документів у різні валідатори.

Мови DSDL все ще не мають підтримки як у XML Schema, і є, певною мірою, реакцією видавців на брак можливостей XML Schema для видавничої справи.

Деякі мови схем не тільки описують структуру певного формату XML-документів, а ще і мають обмежені можливості впливу на обробку документів цього формату.

Як DTD, так і XSD мають цю можливість; наприклад, вони можуть визначати значення для атрибутів «за замовченням». Натомість, як RELAX NG так і Schematron таких можливостей не мають.

Обробка XML-документів

До XML-обробки відносяться форматування, синтаксичний аналіз, редагування, перевірка коректності і перетворення в інші формати.

До традиційних технологій обробки XML-документів належать такі три технології:

1. Написання програм на мові програмування із використанням API SAX.
2. Написання програм на мові програмування із використанням API DOM.
3. Застосування механізму перетворення та фільтра.

До новіших технологій, що почали здобувати поширення останнім часом належать: Активний аналіз. Зв'язування даних.

Простий API для XML (SAX)

Простий програмний інтерфейс для XML (Simple API for XML, SAX) є основаним на подіях інтерфейсом лексичного аналізу. Відповідно до цієї моделі, документ аналізується послідовно, а вміст документа передається на обробники подій аналізатора користувача. SAX є порівняно швидким та легким для реалізації проте складним з точки зору задачі отримання інформації із різних частин XML-документів, оскільки розробник аналізатора мусить дбати про відстеження поточної частини документа. Запропонований SAX підхід краще підходить до ситуацій, коли певний тип інформації завжди обробляється однаково, попри те, в якій частині документа вона знаходиться.

Об'єктна модель документа (DOM)

Об'єктна модель документа (Document Object Model, DOM) є програмним інтерфейсом, який дозволяє здійснювати обхід цілого документа так, наче він є деревом, вузлами якого є об'єкти, що відтворюють зміст документа. Документ DOM може створюватись синтаксичним аналізатором або користувачами (з деякими обмеженнями). Типи даних вузлів DOM-дерев є абстрактними; реалізації мають власні, специфічні для мов програмування типи даних.

4.2 Захист розробленого програмного забезпечення

Tiny Encryption Algorithm (TEA) [1] – блочний алгоритм шифрування типу «Мережі Фейстеля». Алгоритм був розроблений на факультеті комп'ютерних наук Кембриджського університету Девідом Вілером (David Wheeler) і Роджером Нідгемом (Roger Needham) та вперше представлений в 1994 році [2] на симпозиумі зі швидкими алгоритмами шифрування в Льовені (Бельгія).

Шифр не патентований, широко використовується в ряді криптографічних додатків і широкому спектрі апаратного забезпечення, завдяки вкрай низькими вимогами до пам'яті й простоті реалізації. Алгоритм має як програмну реалізацію на різних мовах програмування, так і апаратну реалізацію на інтегральних схемах типу FPGA.

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму TEA, який заснований на бітових операцій з 64-бітним блоком, має 128-бітний ключ шифрування. Стандартна кількість раундів мережі Фейстеля біля 64 (32 циклу), однак, для досягнення найкращої продуктивності або шифрування, число циклів можна варіювати від 8 (16 раундів) до 64 (128 раундів). Мережа Фейстеля несиметрична через використання в якості операції накладення додавання за модулем 2^{32} .

Перевагами шифру є його простота в реалізації, невеликий розмір коду й досить висока швидкість виконання, а також можливість оптимізації виконання на стандартних 32-бітних процесорах, так як в якості основних операцій використовуються операції виключна «АБО» (XOR), побітового зсуву й додавання за модулем 2^{32} . Оскільки алгоритм не використовує таблиць підстановки і раундова функція досить проста, алгоритму потрібно не менше 16 циклів (32 раундів) для досягнення ефективної дифузії, хоча повна дифузія досягається вже через 6 циклів (12 раундів).

Алгоритм має відмінну стійкість до лінійного криптоаналізу і досить гарну до диференціального криптоаналізу. Головним недоліком цього алгоритму

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Також очевидно, що в алгоритмі шифрування TEA немає як такого алгоритму розкладу ключів. Замість цього в непарних раундах використовуються підключі $K [0]$ та $[1]$, у парних – $K [2]$ і $[3]$.

Так як це блочний шифроалгоритм, де довжина блоку 64-біт, а довжина даних може бути не кратна 64-біт, значення всіх байтів, які доповнюють блок до кратності в 64-біт, встановлюється в $0x01$.

КБПЗ_2025

					VKPM-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у магістерської дипломної роботі система гіперконвергентної інфраструктурної платформи.

З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Функціональних кнопок ПЗ.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Верхнього меню: Файл; Аналізатор мережі; Параметри; Довідка.
- Розділу виведення результату роботи системи.

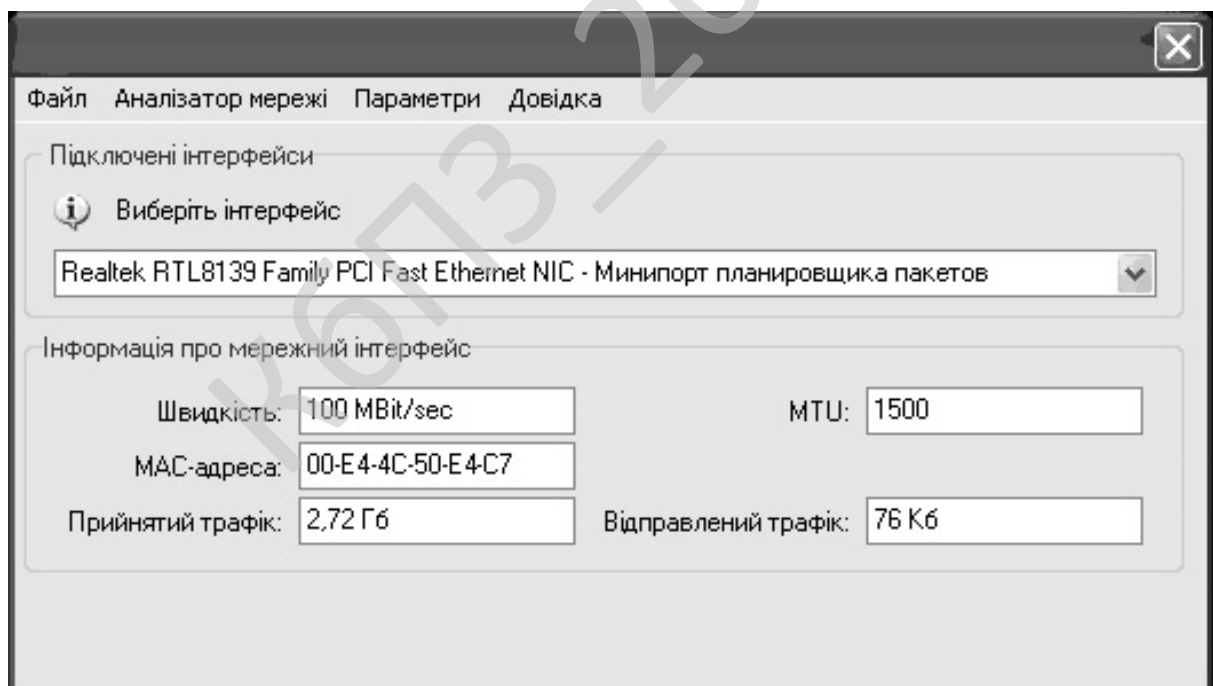


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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.

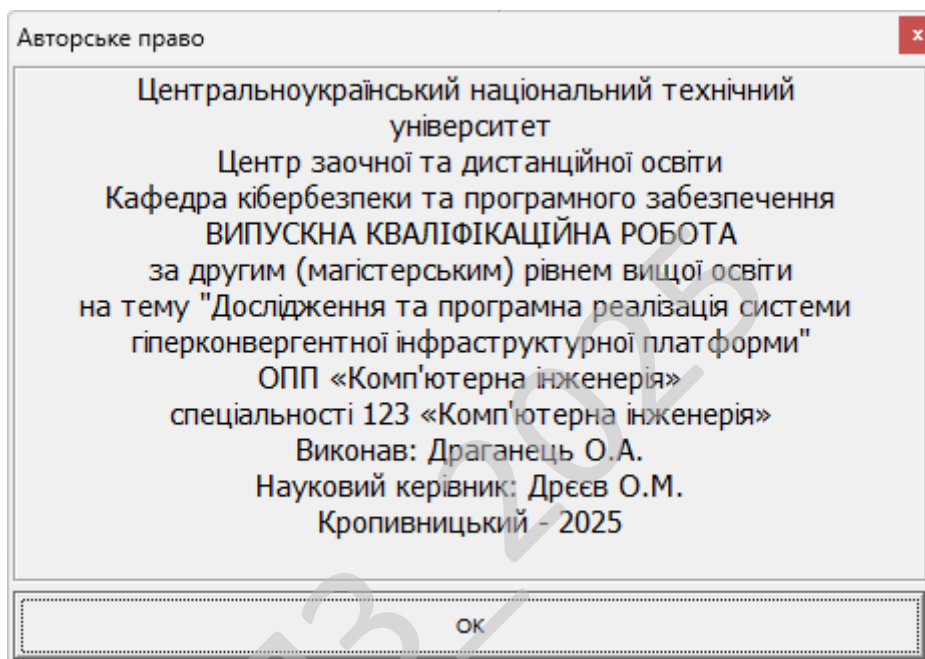


Рисунок 5.2 – Авторське право

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

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

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

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

У цьому випадку формуються тестові варіанти, в яких:

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

- У програмі можуть бути пропущені деякі маршрути.

- Не можна виявити помилки, поява яких залежить від даних.

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

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

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

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

Проводилось тестування чорної скриньки.

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

Ці тести демонструють:

- Як виконуються функції програми.

- Як приймаються вихідні дані.

- Як виробляються результати.

- Як зберігається цілісність зовнішньої інформації.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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 імовірністю є елементами набору ОТ.

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

– Помилки ініціалізації та завершення.

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

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

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

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи гіперконвергентної інфраструктурної платформи.

Метою розробки є дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи.

Об'єктом дослідження є процес гіперконвергентної інфраструктурної платформи.

Предметом дослідження є методи гіперконвергентної інфраструктурної платформи.

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

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

- Удосконалено метод гіперконвергентної інфраструктурної платформи.
- Розроблено вітчизняний продукт гіперконвергентної інфраструктурної платформи, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати такого дослідження насамперед будуть цікавими для ІТ-департаментів великих компаній, що прагнуть зменшити складність управління інфраструктурою та підвищити її продуктивність. Гіперконвергентна інфраструктура поєднує обчислювальні ресурси, зберігання даних та мережеві компоненти в єдину систему, що суттєво спрощує адміністрування. Це особливо актуально для підприємств із власними дата-центрами, банків, телекомунікаційних операторів і державних структур, які працюють із великими масивами даних.

Крім того, такі рішення можуть зацікавити компанії, що займаються хмарними технологіями або послугами з ІТ-аутсорсингу. Для них впровадження НСІ означає зменшення витрат на підтримку інфраструктури клієнтів і підвищення надійності сервісів. Також це перспективний напрям для стартапів, що шукають технологічну базу для швидкого масштабування своїх продуктів без надлишкових витрат на обладнання.

В освітньому та науковому середовищі результати дослідження НСІ можуть бути корисними як приклад практичної реалізації сучасних тенденцій у галузі ІТ-інфраструктури. Студенти, викладачі та дослідники зможуть використовувати такі системи для лабораторних експериментів, тестування хмарних архітектур та вивчення принципів оптимізації ресурсів.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості впровадження гіперконвергентної платформи доцільно залучити експертів у сфері IT-архітектури, економіки підприємства та інформаційної безпеки. Кожен із них може оцінити проєкт за кількома критеріями – рівень інноваційності, потенційна економія витрат, складність інтеграції, масштабованість і рівень ризиків.

Наприклад, за десятибальною шкалою експерти оцінили інноваційність рішення на 9 балів, економічну вигоду – на 8, масштабованість – на 10, а ризики – лише на 3 бали. Після узагальнення результатів середній індекс привабливості склав 8,5 бала, що вказує на високий потенціал впровадження. Такий підхід допомагає об'єктивно оцінити не лише технічну, а й стратегічну цінність проєкту.

Експертна оцінка також дозволяє виявити «вузькі місця» майбутньої реалізації. Наприклад, якщо кілька експертів зазначили, що інтеграція з існуючими системами може бути складною, це дає змогу заздалегідь закласти додаткові ресурси або етапи тестування, зменшуючи майбутні ризики.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості впровадження системи гіперконвергентної інфраструктури доцільно використовувати метод TCO (Total Cost of Ownership) – повної вартості володіння. Він дозволяє врахувати не лише початкову ціну обладнання та програмного забезпечення, а й супутні витрати: навчання персоналу, підтримку, оновлення, електроенергію, охолодження тощо.

Такий метод дає змогу отримати реалістичну картину того, скільки фактично коштує система протягом усього життєвого циклу. Наприклад, традиційна IT-інфраструктура може виглядати дешевшою на старті, але її

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

Розрахунок економічного ефекту демонструє наступне: скорочення витрат на технічне обслуговування – 600 000грн, економія електроенергії – 350 000грн, зменшення витрат на персонал – 960 000грн, економія на ліцензуванні ПЗ – 150 000грн, збільшення продуктивності (еквівалентна вигода від оптимізації бізнес-процесів) – 25% від сукупного ефекту – 515 000грн, загальний економічний ефект за рік – 2 575 000 грн, витрати на впровадження (обладнання + софт + міграція) – 1 800 000 грн, чистий річний економічний ефект – 775 000 грн, термін окупності – 2,3 року, ROI (окупність інвестицій) – 43%.

Додаткові нефінансові вигоди: покращення безперервності бізнесу – мінімізація простоїв і втрати даних завдяки інтегрованим механізмам відновлення, гнучкість масштабування – підприємство може збільшувати ресурси поступово, без масштабних капітальних витрат, підвищення кіберстійкості – єдина архітектура спрощує реалізацію політик безпеки та оновлення, скорочення часу розгортання нових сервісів – з тижнів до годин, що дозволяє бізнесу швидше реагувати на зміни ринку, зниження ризиків морального старіння обладнання – використання віртуалізованого середовища продовжує життєвий цикл інфраструктури.

Впровадження гіперконвергентної інфраструктурної платформи дозволяє підприємству не лише зменшити витрати, але й радикально підвищити ефективність управління IT-ресурсами. Уже протягом перших двох років система повністю окупає себе, а надалі приносить чистий фінансовий ефект понад 700 тис. грн щорічно.

З точки зору стратегічного розвитку, НСІ створює основу для подальшої цифрової трансформації компанії. Вона дає можливість переходити до хмарних архітектур, швидко впроваджувати нові сервіси та гарантувати безперервність бізнесу навіть у разі критичних збоїв.

Таким чином, гіперконвергентна інфраструктура стає не просто технічним рішенням, а довгостроковою інвестицією у стабільність, гнучкість і конкурентоспроможність організації на сучасному ринку.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування проєкту слід почати з аналітичного етапу, коли проводиться оцінка поточного стану ІТ-інфраструктури підприємства. Важливо показати керівництву, які саме проблеми вирішує НСІ: складність масштабування, надмірне енергоспоживання або висока вартість підтримки. Після цього формується демонстраційний стенд або пілотна зона, де потенційні клієнти можуть побачити реальну роботу системи.

Далі варто організувати серію презентацій, вебінарів і технічних воркшопів для ІТ-директорів, системних адміністраторів і бізнес-аналітиків. Саме ці фахівці найчастіше впливають на рішення щодо модернізації інфраструктури. Продумане освітнє просування допоможе зняти побоювання щодо складності переходу на нову платформу.

Фінальним кроком є розробка партнерської програми з виробниками апаратного забезпечення та постачальниками хмарних послуг. Це дозволяє створити екосистему, у якій НСІ стане ключовим елементом, інтегрованим із суміжними технологіями – системами резервного копіювання, моніторингу й безпеки.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту доцільно поєднати прямі продажі великим компаніям із партнерською моделлю для системних інтеграторів і постачальників ІТ-рішень. Це дозволить охопити як корпоративний сегмент, так і середній бізнес. Одним із перспективних напрямів є розповсюдження НСІ як хмарного сервісу (Infrastructure as a Service), що робить технологію доступною навіть для невеликих організацій.

Також можна впровадити гнучку систему ліцензування – наприклад, за кількістю вузлів або обсягом ресурсів, що споживаються. Це підвищить

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

привабливість продукту на ринку, де багато компаній прагнуть уникнути великих одноразових інвестицій. Додатковим фактором оптимізації стане співпраця з навчальними центрами, які готують фахівців з НСІ і сприяють поширенню технології серед майбутніх ІТ-архітекторів.

Не менш важливо забезпечити маркетингову підтримку – створення реальних кейсів і відгуків клієнтів, які продемонструють переваги системи. Це сприятиме формуванню довіри до бренду і стимулюватиме організації до переходу на гіперконвергентну модель.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключовим фактором успіху проєкту є його здатність реально спростити життя ІТ-фахівців і водночас продемонструвати економічний ефект для бізнесу. Якщо система дозволяє скоротити час розгортання серверів із днів до годин, зменшити кількість обладнання та витрати на підтримку, – це вже є вагомим аргументом на користь впровадження.

Іншим важливим аспектом є надійність і масштабованість рішення. Компанії повинні бути впевнені, що НСІ стабільно працюватиме навіть при збільшенні навантаження або під час відмови одного з вузлів. Підтримка виробника, регулярні оновлення й швидке усунення проблем відіграють тут вирішальну роль.

Важливо також забезпечити грамотну комунікацію між технічною та управлінською ланками. Часто саме зрозуміле пояснення переваг технології мовою бізнесу – через цифри, окупність і ефективність – визначає успіх її прийняття. У результаті проєкт стає не просто ІТ-ініціативою, а реальним кроком до цифрової трансформації компанії.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці в галузі ІТ-технологій дозволяє зберегти здоров'я, а іноді й життя працівників, тому є актуальною задачею.

Під час роботи з електронно-обчислювальними машинами (ЕОМ) на користувача у тій чи іншій мірі можуть впливати такі фізичні чинники:

- підвищені рівні змінного електромагнітного і електростатичного полів;
- підвищений рівень статичної електрики;
- підвищений рівень низько енергетичного (м'якого) рентгенівського іонізуючого випромінювання;
- підвищені рівні ультрафіолетового та інфрачервоного випромінювання;
- підвищений вміст позитивних аероіонів у повітрі робочої зони;
- знижений вміст негативних аероіонів;
- аномальний рівень освітленості робочої зони;
- підвищена яскравість фрагментів світлового зображення або світла, що потрапляє у поле зору користувача;
- підвищена нерівномірність розподілу яскравості у полі зору користувача;
- підвищена зовнішня освітленість екрану;
- підвищені пульсації світлового потоку джерел світла або світлового потоку, що випромінюється екраном;
- несприятливий для роботи спектр випромінювання джерел світла;
- підвищена часова нестабільність зображення; мерехтіння екрану;
- зміна яскравості свічення екрана;
- підвищена пряма блискіть, викликана попаданням у поле зору працюючого надмірно яскравого світла різних випромінюючих об'єктів;
- підвищена відбита блискіть, обумовлена наявністю дзеркальних відбиття (відблисків), у тому числі від екрану;

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

- підвищений рівень шуму;
- аномальні температура, вологість та рухливість повітря робочої зони;
- підвищене значення напруги в електричному ланцюзі, замикання якого може статися через тіло людини;
- пожежа.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести також недостатню площу і обсяг виробничого приміщення, недотримання вимог, мікроклімату на робочих місцях.

Праця користувачів ПЕОМ відносять до психічних форм праці з високим ступенем навантаження.

Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження оператора електричним струмом. Саме тому всі працівники, які працюють з персональним комп'ютером, повинні мати першу(або другу) групу допуску з електробезпеки.

Ці шкідливі фактори можуть привести до професійних захворювань.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити:

- безпеку працівників при експлуатації устаткування;
- застосування коштів індивідуальної захисту працівників;
- відповідні вимоги охорони праці, умови праці в кожному робоче місце;
- дотримання режиму праці та відпочинку працівників;
- навчання безпечним методам і прийомам виконання;
- інструктажі з охорони праці;
- організацію контролю над станом умов праці в робочих місць;
- проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – одна з головних задач охорони праці [5].

8.2 Аналіз умов праці на робочому місці фахівця

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,4 м×6,2 м×2,9 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2 м×1,8 м, які виходять на північний захід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита леноліумом, всі стіни пофарбовані світло оранжевого кольору до висоти 2,8 м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1255 мм×845 мм. Висота столів 760 мм. Висота стільців від рівня підлоги становить 430 мм.

Згідно НПАОП 0.00-1.28-10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м² на одне робоче місце. Кратність повітрообміну в приміщенні вузла також регламентується ДСанПіН 3.3.2.007-98 [2], вона повинна становити 20 м³/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні вузла оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5-28-2006 [1] роботу програміста можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення вузла можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому висвітленні), повинен становити 0,5%, освітленість при штучному висвітленні повинна становити 300 лк.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

всього встаткування вузла, включаючи й ксерокс. Це дозволяє зробити висновок про відповідність рівня звуку в приміщенні вимогам нормативних актів.

Ергономічні вимоги до робочого місця працюючого з ВДТ ЕОМ і ПЕОМ нормуються НПАОП 0.00 – 1.28 – 10. Оптимальне положення тіла того, що працює забезпечується відповідною конструкцією робочого місця, а також регуляцією висоти робочої поверхні, сидіння, простори й підставки для ніг. Даного місця програміста не мають регульованих параметрів. Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту дані в таблиці 8.2.

Таблиця 8.2 – Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту

Ріст людини, см	Висота робочої поверхні мм,	Висота простору для ніг, мм	Висота робочого сидіння, мм
175	765(740)	655(600)	450(440)

У дужках зазначені реальні значення параметрів робочого місця; всі вони не відповідають параметрам, зазначеним у стандарті.

Параметри мікроклімату можуть мінятися в широких межах, тоді як необхідною умовою життєдіяльності людини є підтримка сталості температури тіла завдяки властивості терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище.

У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах ДСН 3.3.6.042 – 99 [4] встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення (табл. 8.3).

Таблиця 8.3 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 – 24°C
	Відносна вологість	40 – 60%
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23 – 25°C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

8.3 Розробка заходів з умов поліпшення охорони праці

Провівши аналіз умов праці в розглянутому вище приміщенні, було отримано наступні результати:

- значення мікроклімату в приміщенні не перевищує норму;
- розрахунки розміру робочого місця на одного працівника відповідають нормі;
- рівень шуму в приміщенні не становить вище норми.

З вище перерахованих результатів можна зробити висновок, що основний вплив на продуктивність ІТ-фахівців є його психологічний стан. Тому є доцільним зменшити рівень стресу на робочому місці.

Рекомендовані наступні заходи: За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5-2 м. Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого столу має

регулюватися в межах 680-800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600-1400мм, глибина – 800-1000мм). Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, у глибину (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги не менше ніж 650 мм. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим [7].

8.4 Розрахункова частина

Для захисного штучного заземлення виберемо вертикальні електроди: металевий куток 80x80x8 мм, (згідно з ДСТУ 2251:2018 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною $L=1,6$ м та горизонтальний електрод – металева полоса з перетином 60x5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 8.1).

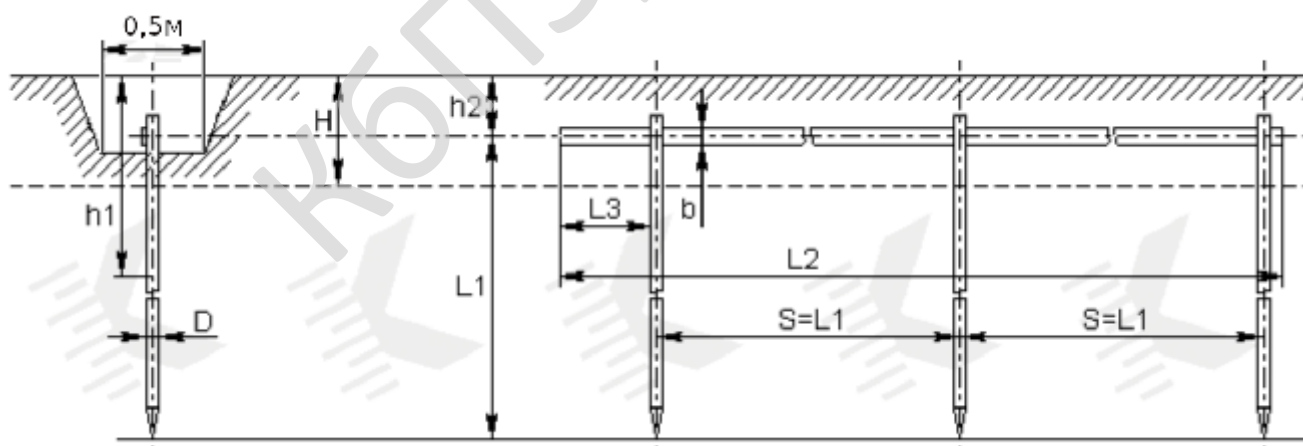


Рисунок 8.1 – Схема штучного заземлення.

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи гіперконвергентної інфраструктурної платформи.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів гіперконвергентної інфраструктурної платформи.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем гіперконвергентної інфраструктурної платформи.
- Досліджена система гіперконвергентної інфраструктурної платформи.
- На основі отриманих результатів досліджень створена програмна реалізація системи гіперконвергентної інфраструктурної платформи.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання гіперконвергентної інфраструктурної платформи.

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

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм TEA.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

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

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Драганець О.А. Дослідження та програмна реалізація системи гіперконвергентної інфраструктурної платформи // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

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

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

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

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

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

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

8. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

9. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

10. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.
11. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.
12. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
13. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
14. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
15. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
16. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.
17. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

18. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

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

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

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

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

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

24. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

25. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

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

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

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

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

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

31. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019,

Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

32. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

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

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

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

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

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

38. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special

Correlation Properties», *CEUR Workshop Proceedings* Volume 2353, *CEUR Workshop Proceedings* 2019, Pages 618-629.

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

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

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

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

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

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

45. Смірнов О.А., 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.

					ВКРМ-123.25.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

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

47. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

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

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

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

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