

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
“ ____ ” _____ 2024 р.

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

КБПЗ - 2024

Виконав здобувач вищої освіти
II курсу, групи КІ-23Мз
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Мороз С.М.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Центр *Заочної та дистанційної освіти*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань 12 *“Інформаційні технології”*
Спеціальність 123 *“Комп’ютерна інженерія”*
Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерна інженерія”*

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 6 » вересня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Морозу Сергію Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи управління гіперЦОД*

2. Керівник роботи *Смірнов Сергій Анатолійович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 21-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи управління гіперЦОД*

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

- | | |
|--|--|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</i> |
| <i>3. Опис і обґрунтування проєктних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> |
| <i>5. Впровадження системи в промислову експлуатацію</i> | |

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

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Мороз С.М. Дослідження та програмна реалізація системи управління гіперЦОД. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

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

Метою розробки є дослідження та програмна реалізація системи управління гіперЦОД.

Об'єктом дослідження є процес управління гіперЦОД.

Предметом дослідження є методи управління гіперЦОД.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи управління гіперЦОД.

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, гіперЦОД

ABSTRACT

Moroz S.M. Research and software implementation of the hyperdatabase management system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software designed for the hyperdatabase management system was developed.

The goal of the development is research and software implementation of the hyperdatabase management system.

The object of the study is the hyperdatabase management process.

The subject of research is hyperdatabase management methods.

Research methods are based on methods of cloud technologies, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the hyperdatabase management system.

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

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

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

The program was developed in the Python environment.

Keywords: computer engineering, hyperdatabase

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	13
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	13
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	24
3.1 Опис функціонування системи	24
3.2 Розробка структурної схеми.....	40
3.3 Розробка функціональної схеми	58
3.4 Розробка діаграми процесів.....	95
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	97
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	97
4.2 Захист розробленого програмного забезпечення.....	112
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	114
6 НАУКОВА НОВИЗНА	120

						ВКРМ-123.24.0004.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Мороз С.М.				Дослідження та програмна реалізація системи управління гіперЦОД	Літ.	Аркуш	Аркушів
Перев.	Смірнов С.А.					М	1	145
Н.контр.	Коваленко А.С.				ЦНТУ КІ-23Мз			
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ ...	121
7.1	Визначення цільової аудиторії кінцевого готового продукту	121
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок .	122
7.3	Вибір методу оцінки вартості ПЗ	123
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	123
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	125
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	125
7.7	Визначення ключових факторів успіху конкретного проєкту.....	127
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	128
8.1	Вступ.....	128
8.2	Аналіз умов праці на робочому місці ІТ-фахівця	129
8.3	Пропозиції щодо підвищення працездатності ІТ-фахівців.....	131
8.4	Розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці	133
8.5	Висновки до розділу.....	136
9	ОСНОВНІ ВИСНОВКИ.....	137
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	139

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

EOM	–	електронно-обчислювальна машина
ATM	–	Asynchronous Transfer Mode – асинхронний спосіб передачі даних
BER	–	Bit Error Rate – імовірність ушкодження одного біта
CBWFQ	–	class based weighted fair queueing
DiffServ	–	Differentiated Services – механізм який залежно від вимог до якості обслуговування записує в IP заголовки пакетів спеціальні мітки
DSCP	–	DiffServ CoSde Point
ICMP	–	Internet Control Message Protocol – міжмережний протокол управляючих повідомлень
ISP	–	Internet Service Provider – провайдер
LLQ	–	low latency queueing
MPLS-TE	–	Multiprotocol Label Switching Traffic Engineering
PQ	–	priority queueing
RIO	–	RED with Input Output
RTT	–	Round Trip Time – час між відправкою запиту та отриманням відповіді
STM	–	Synchronous Transfer Mode – синхронний спосіб передачі даних
QoS	–	Quality of Service – якість обслуговування
WFQ	–	Weighted Fair Queuing – механізм планування пакетних потоків даних
WRED	–	Weighted random early detection – взвішане значення довжини черги, у якості фактора, визначаючого імовірність відкидання пакета

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Загальна ефективність – це саме те, що потрібно в ЦОД. ГіперЦОД концентруються саме на підвищенні ефективності. Економія засобів у перерахуванні на одиницю встаткування дозволяє встановити більше встаткування й забезпечити більше якісні сервіси.

Зміни, що відбуваються в галузі, численні й різноманітні. Можливо, провиною всьому криза, а може бути, наполегливе бажання змінити екосистему – у кожному разі зміни неможливо не помітити.

Зміни торкаються людей, що перетворюють інновації в життя, компанії – як лідерів, так і аутсайдерів – і потреби тих і інших. Потоки даних перемінили обчислювальні операції як типове навантаження для ІТ-устаткування, ПЗ стало відкритим, код абстрагується від процесорної архітектури. Приватні й корпоративні користувачі повстають проти застарілих методів, консервативних вендорів, що втратили свою ефективність бізнес-моделей. Зовсім змінилися ті принципи, відповідно до яких реалізується фундаментальна системна архітектура й взаємодіють між собою процесори.

Отже, якщо ви – системний архітектор, вас не може не захоплювати, які можливості відкриваються для інновацій, для створення нових або трансформації наявних видів бізнесу! Але інновації – це найчастіше реалізація абсолютно нових підходів до тому, як робляться справи. Ще не всі зміни перетворені в життя. Ми тільки на початку шляху.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи управління гіперЦОД.

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

- Огляд існуючих систем управління гіперЦОД.
- Дослідження системи управління гіперЦОД.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Програмна реалізація системи управління гіперЦОД.

Об'єктом дослідження є процес управління гіперЦОД.

Предметом дослідження є методи управління гіперЦОД.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод управління гіперЦОД.

– Розроблено вітчизняний продукт управління гіперЦОД, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі управління гіперЦОД.

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» (2024 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 управління гіперЦОД, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Поява гіперЦОД (це те місце, де зберігаються величезні обсяги даних із соцмереж, пошукових систем і т.п.) і масштаби їхнього розгортання міняють зміст поняття «оптимізація». Одиницею виміру покупки для більшості з нас усе ще є стійка, а для деяких – контейнер (або POD), тоді як у випадку гіперЦОД – це вже окремий ЦОД.

Розглядаючи ситуацію під таким кутом, гіперЦОД прагнуть досягти компромісу між вартістю енергоживлення, площ, використовуваних металевих конструкцій, пропускної здатності мережі, місцем розміщення навантаження й оптимізуємими параметрами додатків. Простіше говорячи, на зміну пошуку локальних оптимумів приходять пошук глобальних оптимумів. Не знаю, як ви, але я ще в університетські роки, займаючись дослідженням операцій, твердо засвоїв, що між двома цими поняттями лежить пропасти. Загальна ефективність – це саме те, що потрібно в ЦОД.

Власники гіперЦОД купують багато встаткування (для шести самих великих, імовірно, здобувається приблизно 10% всієї виробленої техніки), тому, по-перше, вони повинні визначити, що зможуть розробляти самостійно, а по-друге, виробникам прийде серйозно працювати, щоб запропонувати саме те, у чому бідують такі замовники.

Це означає, що якщо раніше рушійною силою інновацій були головним чином OEM-виробники, те тепер їхнє місце зайняли гіперЦОД, причому вони стимулюють їх набагато активніше. Що таке оптимум для ЦОД? Це відношення навантаження до вартості. Причому загального навантаження й загальної вартості. Витратити більше або навіть набагато більше на що-небудь – це

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

нормально, якщо в цілому за один витрачений долар буде виконана більша робота.

От тому самими великими споживачами флеш-накопичувачів у серверах є компанії Facebook, Google і Apple, а за ними з невеликим відставанням впливають деякі інші великі гравці. Ці ЦОД завжди готові швидко надати все, що потрібно, завдяки ефективній роботі флеш-накопичувачів – настільки ефективної, що деякі сервіси доступні безкоштовно.

ГіперЦОД почали публікувати метрики витрат, розкривати відомості про своїх архітектурах (наприклад, OpenCompute) і надавати відкрите ПЗ (проект Nadoor і його похідні). У руслі цих тенденцій Amazon, наприклад, досить точно оцінив вартість своїх сервісів. І вона неймовірно низька.

1.2 Область застосування

Корпоративні клієнти аналізують всі ці цифри, причому досить ретельно. Справа в тому, що багато клієнтів буквально збунтувалися проти старих способів ведення справ в ІТ-галузі – тобто покупки встаткування й виставлення рахунків. OEM-виробники й постачальники послуг, звичайно, сприяють підвищенню ефективності діяльності підприємств, але не в такому ступені, як хотілося б останнім. За допомогою інноваційних рішень вони прагнуть прив'язати клієнта до пропрієтарної архітектури, у той час як гіперЦОД концентруються на підвищенні ефективності. Економія засобів у перерахуванні на одиницю встаткування дозволяє встановити більше встаткування й забезпечити більше якісні сервіси.

Згаданий бунт виражається у двох явищах. Перше помітно у квартальних звітах OEM-виробників і постачальників послуг. IBM збирається продати серію X компанії Lenovo, Dell уживає делістинг, а Oracle намагається знайти новий напрямок для свого бізнесу. При цьому HP анонсує «новий погляд на ІТ»... Друге полягає в тому, що компанії прагнуть у максимально можливому ступені наслідувати гіперЦОД, впроваджуючи архітектури приватних хмар, і, швидше за

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Типова карта SAS для систем зберігання тепер оснащується вісьма портами 12G – а це пропускна здатність на рівні 96G. Чи може бути десять портів 10G у сервера? Навряд чи, скажете ви, і я думаю так само.

Більше прості системи купуються в тих вендорів, які працюють із набагато меншою маржею. Таким чином, витрати на технічне забезпечення й обслуговування істотно знижуються (потрібно обслуговувати меншу кількість пристроїв, а крім того, немає грабіжницьких сервісних контрактів від OEM). До того ж багато дорогих ліцензій на ПЗ замінюються еквівалентами на базі відкритого коду. Чиста економія – 70%. Так що не варто сміятися.

Але, можливо, найважливіша тенденція – це те, що виробники серверів називають дезагрегацією, а фахівці з архітектури ІТ-систем – загальним пулом ресурсів.

По-перше, ціль дезагрегації – це не просто «розчленування» сервера на складові з метою зниження вартості окремих компонентів. Ви однаково здобуваєте стійку – чому б не скомпонувати середовище так, щоб з'єднати подібне з подібним? Кожний компонент має власний життєвий цикл. У процесорів він становить 18 місяців. В DRAM – кілька років. У флеш-пам'яті, швидше за все, три роки. У жорстких дисків – від п'яти до семи. У мережного встаткування – від п'яти до десяти. У джерел живлення – вічність? Чому б не замінити кожний компонент відповідно до його природного життєвого циклу? Чому не привести конструктив у відповідність із вимогами технологій, які він поєднує? Жорстким дискам потрібні стійкі до вібрацій металеві корпуси. Процесорам необхідно гарне охолодження...

По-друге, принцип пула ресурсів дозволяє дійсно ефективно використовувати наявні ресурси. Кожній системі потрібні «жирові» запаси. Що відбудеться, якщо фізична пам'ять буде використовуватися на 100%? Система стане працювати набагато повільніше. Коли в бази даних не вистачає місця для розміщення інформації, з'являється синій екран смерті. У випадку недостатньої пропускної здатності мережі сервери виявляються надлишково

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

укомплектованими для виконання своїх завдань. Зайва пам'ять DRAM, зайва пропускна здатність, зайва ємність флеш-накопичувачів, зайва кількість шпинделів жорстких дисків...Якщо у вас 1000 вузлів, ви викидаєте на вітер терабайти оперативної пам'яті, терабайти флеш-пам'яті, терабайти в секунду пропускної здатності – а на все це понапрасну витрачається енергія. Ще гірше, якщо ви помилитеся в розрахунках і придбаєте сервер з недостатнім обсягом накопичувачів або оперативної пам'яті – з таким оснащенням особливо не розженешся. А тепер представте, що у вас 10 000 або 100 000 вузлів – і жахніться.

Якщо всі ресурси 30–100 серверів звести в єдиний пул, їх можна виділяти відповідно до потреб окремих серверів. Що більше важливо – системи можна конфігурувати логічно, а не фізично, тобто не потрібно максимально точно планувати, яка повинна бути конфігурація, скільки пристроїв буде потрібно і яких саме. У вас є щось начебто «напівфабрикатів», які треба просто розмістити в стійках, після чого їх можна підключити за допомогою конфігураційних сценаріїв і одержати ефективний розподіл ресурсів, яке можна завжди поміняти. Потрібно сховище побільше? Або поменше? Більше продуктивна флеш-пам'ять? Додаткова пропускна здатність мережі? Просто зконфігуруйте на свій смак.

Всі ці міри готують ґрунт для розподілу головної системної пам'яті в загальному пулі ресурсів, що й відбудеться, як тільки будуть готові відповідні рішення наступного покоління – швидше за все, в 2018 році.

Не можна недооцінювати проблеми з масштабною експлуатацією різнорідних платформ. Багато хто гіперЦОД мають до шести платформ. Ви думаєте, нові версії в них встановлюються до того, як припиниться підтримка старих? У дійсності найчастіше одночасно працюють три версії однієї платформи. Тобто, у підсумку використовуються 18 різних платформ, але ж у кожній ще є різні редакції...

Іншими словами, якщо вам потрібно управляти 200 – 400 тис. серверів і здійснювати їхню підтримку без залучення фахівців на місцях, те це може звести з розуму. Організація пула ресурсів значно полегшує експлуатацію.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

імовірно, продовжать програвати). Однак, досить можливо, Intel такі поразки підуть на користь, як це було у випадку AS/400, S/360 або DEC.

Принцип дезагрегації дозволяє гіперЦОД забезпечити роботу сервісів без надмірностей, а переможцем стане той, хто гарантує безперервність поставки. Може бути, у результаті з'явиться нова Intel.

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

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

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

Сучасні тенденції в області ІТ

Сучасні сервісні підходи до побудови й керування інформаційно-обчислювальними середовищами припускають відмова від понять «окремий комп'ютер» або «програма» і, навіть, від поняття «програмно-апаратний комплекс». Замість цього сучасний підхід оперує поняттями «ІТ-сервіс» (інформаційна система) і «ІТ-ресурси». З погляду даного підходу, основні завдання, що коштують перед сучасними ІТ-підрозділами – це:

- побудова ІТ-інфраструктури, орієнтуючись на вимоги бізнес-підрозділів;
- збереження й підвищення якості надаваних ІТ-сервісів;
- забезпечення безпеки функціонування ІТ-сервісів;
- забезпечення безперервності функціонування ІТ-сервісів;
- скорочення загальної вартості володіння ІТ (далі ТСО);

При побудові ІТ-інфраструктури рекомендується використовувати наступні сучасні підходи й технології:

- консолідацію ресурсів;
- віртуалізацію ресурсів;
- сучасну архітектуру додатків;
- модель SaaS (Програмне забезпечення як послуга);
- засоби інтеграції додатків;
- сучасну комунікаційну інфраструктуру.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Консолідація ресурсів

Сучасна ІТ-інфраструктура й, насамперед, комунікаційне середовище створюють передумови для консолідації додатків і даних у сучасних Центрах Обробки Даних (ЦОД). З погляду бізнесу дана модель побудови ІТ-інфраструктури забезпечує більше високу:

– ефективність за рахунок зниження ТСО (насамперед витрат на обслуговування й супровід ІТ систем і зниження капітальних витрат на інфраструктуру, що забезпечує).

– безперервність (доступність) додатків за рахунок більше високого ступеня резервування й відказостійкості програмно-апаратних засобів, застосовуваних у централізованих ЦОД.

При побудові й модернізації ІТ-інфраструктури рекомендується розглядати чотири види консолідації ресурсів:

– Централізація – консолідація географічно розподілених серверів в одному або декількох централізованих дата центрах.

– Консолідація даних – консолідація баз даних і/або пристроїв зберігання для досягнення більше високої доступності й керованості даними.

– Фізична консолідація – об'єднання серверів під керуванням однієї й тією же операційною системою й з подібними додатками, на могутніших системах.

– Консолідація додатків і сховищ даних – розміщення різних додатків на потужних серверах з поділюваними розділами, або на системах віртуалізації.

Практика показує, що загальна вартість володіння консолідованим рішенням значно нижче, ніж вартість володіння його різними компонентами у випадку їхньої роздільної установки.

Віртуалізація ресурсів

Віртуалізація – процес подання набору обчислювальних ресурсів, або їхнього логічного об'єднання, що дає які-небудь переваги перед оригінальною конфігурацією. Це новий, «віртуальний» погляд на ресурси, не обмежених реалізацією, географічним положенням або фізичною конфігурацією складових

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

частин. Звичайно віртуалізовані ресурси містять у собі обчислювальні потужності й сховище даних.

Віртуалізація – це загальний термін, що охоплює абстракцію ресурсів для багатьох аспектів обчислень. Типи віртуалізації приводяться нижче:

– Програмна віртуалізація: віртуалізація на рівні операційної системи, що дозволяє безлічі віртуалізованих середовищ виконуватися на одному екземплярі операційної системи.

– Апаратна віртуалізація: віртуалізація апаратних ресурсів, що дозволяє виконуватися операційним системам виконуватися в оточеннях, що приховують від них фізичні характеристики платформи-носія.

Області застосування віртуалізації:

– Віртуалізація на рівні операційної системи: віртуалізує фізичний сервер на рівні ОС, дозволяючи запускати ізольовані й безпечні віртуальні сервери на одному фізичному сервері. Ця технологія не дозволяє запускати ОС із ядрами, відмінними від типу ядра базової ОС. При віртуалізації на рівні операційної системи не існує окремого шару гіпервізора. Замість цього сама хостова операційна система відповідає за поділ апаратних ресурсів між декількома віртуальними серверами й підтримку їхньої незалежності друг від друга.

– Віртуальні машини: оточення, що представляється для «гостьової» операційної системи, як апаратне. Однак насправді це програмне оточення, що емулюється програмним забезпеченням хостової системи. Ця емуляція повинна бути досить надійної, щоб драйвери гостьової системи могли стабільно працювати. При використанні паравіртуалізації, віртуальна машина не емулює апаратне забезпечення, а, замість цього, пропонує використовувати спеціальний інтерфейс прикладного програмування (API).

– Віртуалізація ресурсів: поділ одного фізичного сервера на кілька частин, кожна з яких видна для власника як окремий сервер. Не є технологією віртуальних машин, здійснюється на рівні ядра ОС.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

– Віртуалізація додатків: процес використання додатка перетвореного з потребуючої установки в ОС у не потребуючий цього. Для віртуалізації додатків програмне забезпечення віртуалізатора визначає при установці віртуалізуемого додатка, які потрібні компоненти ОС і їх емулює, таким чином, створюється необхідне спеціалізоване середовище для конкретно цього віртуалізуемого додатка й, тим самим, забезпечується ізольованість роботи цього додатка.

Поняття «віртуальні машини» припускає поділ на:

– Віртуалізацію серверів: розміщення декількох логічних серверів у рамках одна фізичного (консолідація). об'єднання декількох фізичних серверів в один логічний для рішення певного завдання.

– Віртуалізацію робітників станцій: використання на робочому місці «тонких» клієнтів, що дозволяють виконувати всі необхідні користувачеві завдання на сервері в окремій для кожного клієнта віртуалізованої операційній системі.

Говорячи про серверний віртуалізації, основні передумови застосування технологій наступні:

– неухильне підвищення потужності одиничного сервера, найчастіше перевищуючої потреби конкретного додатка;

– велика кількість наслідуваних додатків на застаріваючих операційних системах і апаратних платформах;

– тенденції до консолідації ІТ-інфраструктури.

Рекомендується використовувати технології й засоби віртуалізації ІТ-ресурсів при побудові ІТ-інфраструктури.

Логічний розподіл обчислювальних комплексів

Технологія апаратних і програмних розділів (partitioning) – це архітектурний підхід до ІТ-інфраструктури, що дозволяє віртуалізувати апаратні ресурси й зробити їх доступними для безлічі незалежних операційних середовищ. Споконвічно розроблена для mainframe, технологія дозволяє розділити один сервер на трохи повністю незалежних апаратні або програмних віртуальних

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

серверів або логічних розділів. Технологія з невеликими розходженнями підтримується провідними виробниками ІТ-устаткування. Технологія логічного розподілу обчислювальних комплексів у сполученні із планами відновлення після збоїв і відновлення після катастроф (DRS – Disaster Recovery System і DRP – Disaster Recovery Plan) і з використанням двох рознесених ЦОД (основн і резервного), створюють основу катастрофостійкої інфраструктури таким чином, що додаток і розділи із критичними додатками, виконувані в основному ЦОД, можуть перемикатися на резервний ЦОД з мінімальними втратами й прозоро для користувачів додатків.

Сучасна архітектура додатків

Рекомендується віддавати перевагу ІТ-рішенням, що мають сучасну, сервіс-орієнтовану багаторівневу архітектуру.

Сервіс-орієнтована архітектура – SOA

Найбільш перспективна архітектура побудови додатків на теперішній час – Service Oriented Architecture (SOA – сервіс-орієнтована архітектура). Основне завдання SOA полегшити інтеграцію додатків – як нових програмних рішень, так і систем попередніх поколінь. Архітектура SOA незалежна від мов програмування, платформ або протокольних специфікацій, за допомогою яких сервіси розробляються, а також від того, де й за допомогою чого вони розгорнуті. Практично архітектура SOA вимагає наявності не тільки сервісів, але й засобів, за допомогою яких ці сервіси можуть бути виявлені й підключені, а також безлічі компонентів, таких, як: сервери додатків, що зв'язують ПЗ, репозиторій і навіть спеціалізовані пакети централізованого керування SOA.

Багаторівнева архітектура клієнт-сервер

Клієнт-сервер – обчислювальна або мережна архітектура, у якій завдання або мережне навантаження розподілені між постачальниками послуг (сервісів), названими серверами, і замовниками послуг, названими клієнтами. Нерідко клієнти й сервери взаємодіють через комп'ютерну мережу й можуть бути як різними фізичними пристроями, так і програмним забезпеченням.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

У цей час більшість інформаційних систем проектуються з використанням трирівневої архітектури «клієнт-сервер», що представляє інформаційну систему у вигляді сукупності трьох компонентів: сервера баз даних, сервера додатків, відповідального за виконання логіки додатка, і клієнтського додатка або клієнтського інтерфейсу («тонкий клієнт»). Основними перевагами виділення логіки додатка в окрему складову є можливість повторного використання коду, легкість модифікації централізовано розгорнутих компонентів, підвищення продуктивності використовуваного сервера бази даних, можливість масштабування системи в цілому й незалежність системи від фізичного розташування бази даних. Крім того, системи, побудовані в трирівневій архітектурі, істотно простіше й дешевше в експлуатації, тому що всі виправлення й конфігураційні налаштування вносяться централізовано.

Складові трирівневої архітектури:

- рівень подання (реалізуючий функції введення й відображення даних);
- прикладний рівень (реалізуючий універсальні сервіси, а також функції, специфічні для певної предметної області);
- рівень доступу до інформаційних ресурсів (реалізуючий фундаментальні функції зберігання й керування інформаційно-обчислювальними ресурсами).

Модель SaaS

Програмне забезпечення як послуга (*software as a service*, сокр. *SaaS*) – модель використання програмного забезпечення, при якій постачальник розробляє веб-додаток і самостійно управляє їм, надаючи користувачам доступ до програмного забезпечення через мережу. Основна перевага моделі SaaS для користувача складається у відсутності витрат, пов'язаних з установкою, відновленням і підтримкою працездатності встаткування й працюючого на ньому програмного забезпечення. Використання SaaS переважніше як більше дешева й проста альтернатива внутрішнім інформаційним ресурсам.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Засоби інтеграції додатків (middleware)

Для побудови інтегрованої ІТ інфраструктури, особливо при використанні різних програмно-апаратних середовищ, рекомендується використовувати спеціалізовані програмні засоби – Інтеграційні платформи. Інтеграційна платформа повинна забезпечувати доступ у реальному масштабі часу до різних інформаційних ресурсів, обмін і синхронізацію даних між ними, автоматично, за заданими правилами й розкладу – підтримку єдиного стандарту обміну інформацією між додатками, незалежно від платформ, на яких вони існують.

Базові функціональні можливості повинні містити в собі засобу перевірки коректності, очищення, об'єднання структурованої й неструктурованої даних, реплікації даних і публікації інформації про події, а також засобу корпоративного пошуку.

Продукти такого типу, як правило, будуються на принципах сервіс-орієнтованої архітектури й призначені для рішення завдань забезпечення вірогідності інформації в масштабі організації, контролю якості даних, їхнього перетворення, переміщення й об'єднання, а також керування метаданими.

Застосування даних технологій дозволяє, за допомогою відповідних сервісів, надаваних додаткам, процесам і окремим користувачам, швидко зіставити, погодити й об'єднати розрізнені дані, що надходять із різних джерел.

Сучасна комунікаційна інфраструктура

Для побудови мереж передачі даних рекомендується застосовувати технології побудови віртуальної приватної мережі на базі MPLS/VPN, з використанням послуг, надаваних великими телекомунікаційними операторами національного масштабу. При необхідності, резервування каналів здійснюється телекомунікаційними операторами, у тому числі, з використанням супутникового зв'язку.

Побудова внутрішньої мережної інфраструктури рекомендується робити на базі сучасних принципів побудови мережі, що забезпечують гарантовану якість мережних послуг (QoS).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Архітектура таких мереж складається із чотирьох основних компонентів, а саме:

- Інтелектуальна мережна інфраструктура на базі протоколу IP.
- Інтелектуальні клієнтські місця з підтримкою протоколу IP.
- Службові серверні додатки.
- Сучасні користувальницькі додатки.

Основа архітектури – її розподілена природа, завдяки якій система легко масштабується. За рахунок цього, мережею на базі даної архітектури можна охопити один будинок або кілька вартим рядом будинків, об'єднаних високошвидкісною локальною мережею, надати в мережі сервіси телефонії, відео- і даних для користувачів віддалених офісів і підрозділів, об'єднаних IP мережею.

Захист даних у мережах передачі даних, як у внутрішніх, так і в зовнішніх, повинна будуватися відповідно до "Політики інформаційної безпеки інформації".

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Python організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Python на самому Python, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

Python – мова програмування, що активно розвивається, нові версії (з додаванням/зміною мовних властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхню роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

При цьому, на відміну від багатьох портуємих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Більше того, існує спеціальна версія Python для віртуальної машини Java – Jython, що дозволяє інтерпретаторові виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python й навіть бути написаними на Python. Також кілька проектів забезпечують інтеграцію із платформою Microsoft .NET, основні з яких – IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто тип змінної визначається тільки під час виконання. Тому замість «присвоювання значення змінної» краще говорити про «зв'язування значення з деяким ім'ям». У Python є убудовані типи: бульові, рядки, Unicode-рядки, цілі числа довільної точності, числа із плаваючою комою, комплексні числа й деякі інші. З колекцій Python підтримує кортежі

(*tuples*), списки, словники (асоціативні масиви) і, починаючи з версії 2.4, безлічі. Всі значення в Python є об'єктами, у тому числі функції, методи, модулі, класи.

Додати новий тип можна або написавши клас (*class*), або визначивши новий тип у модулі розширення (наприклад, написаному мовою C). Система класів підтримує спадкування (одиначне й множинне) і метапрограмування. Можливе спадкування від більшості убудованих типів і типів розширень.

Всі об'єкти діляться на посилальні й атомарні. До атомарного ставляться *int*, *long*, *complex* і деякі інші. При присвоюванні атомарних об'єктів копіюється їхнє значення, у той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують те саме значення. Посилальні об'єкти бувають змінювані й незмінні. Наприклад, рядки й кортежі є незмінними, а списки, словники й багато інших об'єктів – змінюваними. Кортеж у Python є, по суті, незмінним списком. У багатьох випадках кортежі працюють швидше списків, тому якщо ви не плануєте змінювати послідовність, то краще використовувати саме їх.

Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Python програм легко читаємий.

Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якість: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи управління гіперЦОД.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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 методика 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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Рекомендації з керування ІТ інфраструктурою

Оцінка необхідності змін в ІТ-інфраструктурі

При організації діяльності ІТ-служби Замовника рекомендується використовувати сервісний підхід, основи якого викладені в «Стандарті надання послуг в області інформаційних технологій». Відповідно до сервісного підходу діяльність ІТ-служби – це надання якісних і надійних послуг, що задовольняють вимогам інших підрозділів компанії. За допомогою сервісного підходу здійснюється:

- організація ІТ відповідно до вимог бізнес-процесів;
- використання описів ІТ-послуг (сервісів) мовою, зрозумілому їхнім користувачам;
- керування ІТ у термінах надаваних послуг (сервісів);
- керування витратами на використання надаваних послуг (сервісів);
- вимірність результатів інвестицій в ІТ і оптимізація сукупної вартості володіння (ТСО) послугами (сервісами).

При рішенні завдань керування ІТ-інфраструктурою важливе значення мають процеси, пов'язані зі змінами ІТ-інфраструктури. Технічні рішення, пов'язані зі змінами ІТ-інфраструктури, повинні бути обґрунтовані, у тому числі, з урахуванням оцінки їхнього впливу на вартість і якість послуг, надаваних ІТ-службою, і реалізовані з урахуванням наступних рекомендацій по внесенню змін і плануванню ІТ-інфраструктури.

Рекомендації із проведення змін в ІТ інфраструктурі

Процес по проведенню змін в ІТ-інфраструктурі – це процес використання стандартизованих методів і процедур для ефективного й своєчасного проведення

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

змін в ІТ-інфраструктурі з мінімальними негативними наслідками для бізнес-процесів.

При зміні ІТ-інфраструктури необхідно враховувати наступні вимоги:

- Загальні вимоги до тестування й приймання змін.
- Вимог «розумного консерватизму» при впровадженні нових версій ІТ-компонентів.
- Вимог автоматизації тиражування відновлень ПЗ.
- Обов'язкове документування внесених змін.

Загальні вимоги до тестування й приймання змін

Процес тестування й приймання змін ІТ-інфраструктури повинен задовольняти наступним загальним вимогам:

– Рішення про проведення змін повинні прийматися Службою замовника за узгодженням з функціональними замовниками. Служба замовника вправі розробити спрощений порядок прийняття рішень про проведення некритичних для бізнес-процесів змін.

– Перед введенням змін в експлуатацію необхідно розробити й протестувати план внесення змін з обов'язковою вказівкою контрольних точок ухвалення рішення про відмову від внесених змін і поверненні системи в попередній стан. Тестування плану внесення змін необхідно робити в середовищі, аналогічній виробничій.

– Тестування й приймання змін необхідно робити в середовищі, аналогічній виробничій.

– Введення змін в експлуатацію випереджається резервним копіюванням всіх елементів системи, які підверглись зміні.

Вимоги «розумного консерватизму»

При впровадженні, модернізації, відновленні ІТ-компонентів повинні бути дотримані наступні загальні вимоги «розумного консерватизму»:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

– впровадження нових версій ПЗ повинне мати обґрунтовані переваги перед використовуваними версіями ПЗ й не повинне погіршувати поточного стану справ;

– для критично важливих бізнес-процесів неприпустиме впровадження останніх версій промислового ПЗ й/або його компонентів, за винятком використовуваних у схожому промисловому середовищі більше 2 років і, що мають при цьому не менш 2 коригувальних відновлень від виробника (релізів, патчів, сервіс-паків, відновлень ПЗ й т.п.);

– неприпустиме використання тестових версій ІТ-компонентів (альфа, бета версії ПЗ й т.п.) у режимі промислової експлуатації;

– при плануванні впровадження нових версій ПЗ рекомендується враховувати довгострокові плани виробника по випуску нових версій/релізів;

– перед впровадженням нових версій ПЗ в промислову експлуатацію обов'язково ретельне тестування функціонування відповідної системи, з висновками відповідних ключових фахівців.

Автоматизація тиражування відновлень ПЗ

Автоматизація тиражування відновлень ПЗ повинна виконуватися з обліком наступних загальних вимог і рекомендацій:

– при виборі ПЗ, при інших рівних функціональних характеристиках, перевага варто віддавати системам, що мають служби централізованого керування й відновлення;

– створення й впровадження рішення по автоматизації тиражування відновлень ПЗ повинне вироблятися в рамках окремих проектів ІТ-служби й з використанням власної серверної системи тиражування відновлень; Відновлення ПЗ безпосередньо з Інтернет-порталу виробника не допускається;

– функціональні можливості автоматизованої системи тиражування відновлень повинні дозволяти реалізацію повного й спрощеного циклів керування змінами, проводити автоматичну перевірку (аудит) тиражування відновлень;

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

– тиражування відновлень в області безпеки критичних для підприємства систем (відновлення операційних систем, ПЗ з безпосереднім доступом у зовнішні мережі / Інтернет, антивірусні системи й т.п.) підлягає обов'язковій автоматизації;

– тиражування відновлень в області безпеки й функціональних відновлень на робочі станції й інфраструктурні ІТ-компоненти (домен – контролери, DNS / DHCP – сервера й т.п.) у частині системного й офісного ПЗ підлягає обов'язковій автоматизації;

– відновлення інфраструктурного ПЗ (домен-контролери, DNS/DHCP/mail-сервера), що включає зміну функціональності (крім відновлень в області безпеки), підвищення major-версії необхідно проводити під контролем відповідного фахівця, попередньо провівши тестування в тестовому середовищі, аналогічній виробничій;

– рекомендується автоматизація тиражування відновлень клієнтської частини бізнес-додатків;

– перед масштабним тиражуванням відновлень необхідно провести тестування в локальному тестовому середовищі.

Рекомендації із планування ІТ-інфраструктури

При плануванні ІТ-інфраструктури повинен бути забезпечений економічно обґрунтований рівень відповідності ресурсів ІТ-інфраструктури поточній і майбутній потребам споживачів. Для ефективного планування ІТ-інфраструктури необхідно враховувати прогноз розвитку основної діяльності споживача й технічного розвитку ІТ. Тому, для кожного споживача, важливе значення має вибір об'єктів планування для різних ІТ-додатків і визначення базової методики розрахунку продуктивності й обсягу збереженої інформації для ІТ-систем.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Рекомендації з вибору обріїв планування для ІТ-додатків

Вибір обріїв планування ІТ-додатків повинен визначатися обрієм планування споживача. При цьому обрій планування ІТ-інфраструктури повинен бути менше, ніж обрій планування ІТ-додатків.

Рекомендується виходити з наступних обріїв планування ІТ-додатків:

- 8-9 років для додатків класу ERP і додатків керування ІТ, критичних для бізнесу;
- 5-6 років для бізнес-додатків і систем операційного керування ІТ;
- 3-5 років для офісних і системних додатків.

При цьому для відповідних елементів ІТ інфраструктури рекомендується встановити наступні обрії планування :

- 5-6 років для додатків класу ERP і додатків керування ІТ, критичних для бізнесу;
- 3-4 роки для бізнес-додатків і систем операційного керування ІТ;
- 2-3 роки для офісних і системних додатків.

Рекомендації з розрахунку продуктивності й обсягу збереженої інформації для ІТ-систем (масштабування ІТ-систем)

Розрахунок продуктивності ІТ-систем повинен вироблятися в термінах ІТ-послуг, виходячи із планованих строків використання (обрію планування), обсягів, значень показників рівня надання й продуктивності послуг.

При розрахунку продуктивності підлягають обліку навантажувальні можливості всіх задіяних у наданні ІТ-послуг компонентів: ІТ-послуг, ІТ-систем, компонентів інфраструктури й систем інформаційної безпеки, включаючи клієнтські робітники станції.

При розрахунку продуктивності ІТ-систем обов'язковий облік як середніх, так і пікових показників навантаження. Рекомендується для розрахунку продуктивності використовувати надаваний виробником інструментарій – навантажувальні криві й розроблені кращі методики (bestpractices).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

У загальному випадку, у ході розрахунку рекомендується проводити оцінку завантаження по наступних компонентах:

- сервер додатків;
- сервер баз даних;
- серверні засоби інформаційної безпеки (антивірусне забезпечення, програмний міжмережний екран, криптографічні системи й т.п.);
- компоненти моніторингу (агенти й т.п.);
- системне ПЗ;
- апаратна платформа (процесори, оперативна пам'ять, дискова підсистема, мережний інтерфейс);
- мережі зберігання даних (SAN, NAS);
- мережі передачі даних, включаючи активні пристрої (міжмережні екрани, маршрутизатори й т.п.);
- мережні сервіси (сервіси каталогу, DNS, DHCP і т.п.);
- клієнтські робочі станції в апаратній і програмній частині.

Для оцінки обсягу збереженої інформації рекомендується використовувати власні статистичні дані по обсягах і динаміці зміни даних за тривалі (від 1 місяця) періоди й максимальні значення в пікові періоди. При відсутності власних даних припустиме використання статистики схожих підприємств або експертних оцінок. При впровадженні нових сервісів варто дотримуватися інформації, що вказується виробником, як в області початкових вимог, так і в прогнозованих обсягах збільшення інформації. Останній показник варто коректувати в ході наступної експлуатації систем.

При оцінці обсягу збереженої інформації обов'язковий облік не тільки корисного обсягу, але й обсягів системної й службової інформації (системи шифрування, файли логування, файли журналів транзакцій і т.п.).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Каталогізація й класифікація елементів ІТ-інфраструктури

Типізація елементів ІТ-інфраструктури

Уніфікація й типізація елементів ІТ-інфраструктури здатна значно знизити витрати й полегшити впровадження нових ІТ-рішень, знизити витрати на підготовку ІТ-персоналу, спростити супровід і обслуговування цих рішень у майбутньому й, в остаточному підсумку, значно знизити загальну вартість володіння ІТ-інфраструктурою в цілому.

Під типізацією розуміється зниження номенклатури й підвищення якості ІТ-елементів і конфігурацій, впроваджуваних в органах державної влади.

Дана технічна політика висуває вимоги до наступних категорій ІТ інфраструктури:

- робочим місцям користувачів (ПК, периферійне встаткування);
- мультисервісної мережі (корпоративна мережа, зовнішні канали зв'язку);
- прикладного ПЗ;
- інфраструктурі центрів обробки даних (системи зберігання й резервування, сервери).

Класифікація за рівнем використання

При побудові сучасної ІТ-інфраструктури для визначення місця установки (розміщення) ІТ-рішення необхідно провести класифікацію даного рішення залежно від сукупності наступних параметрів: состава користувачів даної системи, ступеня агрегації інформації, зберігання й обробки даних, розв'язуваних задач, рівня складності й т.д.

З погляду даної класифікації рекомендується виділити наступні класи або рівні інфраструктури для розміщення ІТ-систем:

- Центр обробки даних – ЦОД I рівня, (рівень Уряду) – забезпечує централізоване зберігання й обробку даних на рівні Уряду. Ресурси даного ЦОД використовуються для консолідації інформації з нижчележачих рівнів,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

забезпечення інформаційної взаємодії органів державної влади й надання ІТ-сервісів всім організаціям, фінансованим з республіканського бюджету.

– Центр обробки даних – ЦОД II рівня, основний центр зберігання й обробки даних у рамках одного органа державної влади.

Класифікація за рівнем необхідної безперервності обслуговування й важливості для бізнесу

Багато критичних управлінських і технологічних процесів опираються на комп'ютерні системи обробки й зберігання даних і не можуть функціонувати без їхнього використання. Тому, забезпечення безперервності обслуговування й доступності ІТ-рішень є найважливішим показником безперервності державного керування в цілому, і важливим фактором, що класифікує, для елементів ІТ-інфраструктури. Виходячи із пропонованих вимог до надійності окремих елементів і конфігурацій ІТ-систем у цілому і їхньому відновленні після збоїв і відмови встаткування, ПЗ або інфраструктурних елементів, сучасні ІТ-технології надають різні архітектурні й конфігураційні рішення, що забезпечують дані вимоги. З погляду забезпечення безперервності обслуговування управлінських і технологічних користувачів і процесів, а також вимог до відказостійкості, можна запропонувати наступну класифікацію ІТ-рішень:

– **Mission Critical-системи**, що працюють у режимі «бойового чергування». До таких систем ставляться: гостро критичні з погляду державного керування або зовнішніх факторів – наприклад екології, додатки, а також технологічні додатки, що працюють у режимі реального часу. Вихід з ладу цих систем спричиняє непоправні втрати для керування, у т.ч. загрозу життю й здоров'ю персоналу й населення. Рекомендований час відновлення подібних систем після відмови менш 10 хвилин. Для таких систем повинні використовуватися спеціалізовані серверні платформи й інфраструктурні рівні з повним багаторазовим резервуванням всіх компонентів, у тому числі з використанням резервних віддалених ЦОД.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– **Business Critical-системи**, критичні для керування, з режимом роботи 24x7x365. Вихід з ладу цих систем спричиняє значні бізнес втрати для органів державної влади. Рекомендований час відновлення подібних систем після відмови менш 2 годин. Для таких систем повинні використовуватися кластерні рішення й інфраструктурні рівні із частковим резервуванням використовуваних інфраструктурних компонентів.

– **Business Operational** – звичайні бізнес-додатки – системи, не потребує роботи в реальному часі, з режимом роботи 8x5. Рекомендований час відновлення подібних систем після відмови 4-6 години. Для таких систем рекомендується використовувати резервування зберігання даних і електроживлення.

– **Office Production** – не критичні для управління додатка, персональні дані. Рекомендований час відновлення подібних систем після відмови 1-2 робочих дня.

Необхідно враховувати, що загальна безперервність і відказостійкість ІТ-конфігурацій визначається відповідною безперервністю й відказостійкістю її окремих елементів: апаратних, програмних засобів і інфраструктури, необхідної для її успішного функціонування – каналів зв'язку, системи електроживлення й т.д. і, в остаточному підсумку, залежить від рівня безперервності й відказостійкості його найслабшого компонента (принцип «слабкої ланки»). Класифікація систем з погляду забезпечення безперервності й відказостійкості повинна бути одним з вирішальних факторів при виборі рівня інфраструктури (ЦОД) для розміщення ІТ-систем.

Принципи створення КРК

Необхідно розробити й впровадити каталог рекомендованих конфігурацій (КРК) для побудови ІТ-рішень. Даний каталог повинен містити перелік типових елементів ІТ-інфраструктури, рекомендованих до використання в органах державної влади, державних установ.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Основна мета створення КРК – провести уніфікацію використовуваного встаткування й програмного забезпечення й забезпечити цілісність і керованість ІТ-інфраструктури органів державної влади, державних установ.

При побудові й розвитку своєї ІТ-інфраструктури всі органи державної влади, державні установи повинні закупаувати й впроваджувати тільки внесені в каталог ІТ-конфігурації. Закупівля конфігурацій, які не входять у даний список, можлива в тільки виді виключення.

Уніфікація ІТ-рішень, використовуваних у рамках конкретного органа державної влади дозволить домогтися зниження загального ТСО, що має на увазі зниження витрат на закупівлі, впровадження й експлуатацію елементів ІТ-інфраструктури.

Каталог створюється в кожному органі державної влади й повинен містити наступний мінімальний набір даних про рекомендовану конфігурацію:

- назва конфігурації;
- клас об'єкта (функціональний розділ каталогу) відповідно до «Класифікатора об'єктів»;
- клас конфігурації за рівнем використання;
- клас конфігурації за рівнем безперервності;
- набір апаратних засобів, що рекомендується, для даної конфігурації;
- набір програмних засобів, що рекомендується, для даної конфігурації.
- Кількість уніфікованого встаткування й ПЗ в кожному функціональному розділі каталогу повинне бути мінімально.
- Рекомендується використовувати стандартне встаткування й ПЗ с'єбебе, що зарекомендував, на ринку виробника в даній області, що постійно розвиває й удосконалює свій модельний ряд.
- Каталог повинен регулярно (не рідше чим раз у рік) переглядатися й обновлятися.
- Рекомендується використовувати складне апаратно-програмне забезпечення різного призначення того самого виробника. Таке рішення спрощує

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

керування ІТ-інфраструктурою, дозволяє знизити експлуатаційні витрати, а також вартість знову апаратно-програмного забезпечення, що здобувається.

– Для масових, стандартних ІТ-конфігурацій і їхніх елементів рекомендується включати в каталог рішення не менш чим двох альтернативних виробників.

Вимоги до постачальників, виробникам і проведенню конкурсів

У справжньому розділі викладені основні вимоги, що стосуються вибору постачальників програмно-апаратних комплексів і послуг, виробників програмно-апаратних комплексів, а так само системних інтеграторів, що здійснюють розгортання ІТ-інфраструктури.

Органи державної влади вправі при виборі постачальників і виробників орієнтуватися на більше високі вимоги, чим це передбачено в справжньому документі.

Загальні вимоги

До постачальників і виробників пред'являються наступні загальні вимоги:

– Діяльність постачальників і виробників повинна бути ліцензована, якщо таке передбачає Українське законодавство.

– При підведенні підсумків конкурсу, за інших рівних умов, перевагу мають ті компанії, виробництво яких відповідає вимогам стандарту системи менеджменту якості ISO 9001:2000. Причому в додатку до сертифіката повинна бути зазначена саме та діяльність, на надання якої претендує дана компанія.

– За останні три роки учасник конкурсу повинен успішно завершити два аналогічних контракти на поставку послуг, якщо він претендує на постачальника послуг, або розробку, інсталяцію, забезпечення технічної підтримки для ІТ-систем, якщо він претендує на поставку програмно-апаратного забезпечення, аналогічних конкурсній пропозиції по параметрах і порівнянних по масштабі.

– Мінімально прийнятний середньорічний оберт постачальника за останні три роки не повинен бути менше, ніж сума договору на поставку послуг або встаткування, помножена на 5.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Ключові керівники постачальника повинні мати досвід успішної реалізації, як мінімум, двох контрактів порівнянного масштабу, з яких мінімум один контракт реалізований в Україні.

Вимоги до постачальників і виробників устаткування

До постачальників і виробників програмно-апаратного забезпечення пред'являються наступні вимоги:

– Виробник апаратного забезпечення повинен мати сертифікат, виданий визнаним органом по сертифікації, на відповідність своєї системи менеджменту якості вимогам ISO 9001:2000.

– Перевага повинне віддаватися тим постачальникам апаратного забезпечення які мають сертифікат, виданий визнаним органом по сертифікації, на відповідність своєї системи менеджменту якості вимогам ISO 9001:2000.

– Апаратна платформа й програмне забезпечення повинні бути стандартизовані й сертифіковані на відповідність стандартам, що вважаються загальноприйнятими в предметній області даного програмно-апаратного забезпечення, мати гнучку й масштабовану архітектуру.

– Пропоноване постачальником устаткування повинне бути комерційно доступно протягом не менш трьох місяців, а програмне забезпечення не менш шести місяців.

– Постачальники складного програмно-апаратного забезпечення повинні вчасно одержувати сертифікати відповідності на всю гаму встаткування, що поставляється, і ПЗ відповідно до діючих технічних вимог у повному обсязі, а також обновляти сертифікати з появою нових версій програмного й апаратного забезпечення поставляється продукції, що.

– У випадку поставки встаткування, постачальник повинен мати власний склад.

– Постачальник повинен бути авторизований виробником на той перелік послуг, які постачальник буде робити замовникові. При цьому перевага повинне

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

віддаватися постачальникам, що мають більше високий статус (рівень) відносин з виробником.

– Постачальник повинен здійснювати страхування вантажу при доставці.
– Постачальник повинен мати сертифікований технічний персонал на ту предметну область, у якій постачальник буде робити замовникові послуги.

– Перевага повинне віддаватися виробникам, що поставляють устаткування й ПЗ, адаптоване для використання в Україні, якщо така адаптація можлива.

– Перевага повинне віддаватися постачальникам, що надають документацію на встаткування й ПЗ українською мовою й на електронних носіях (в електронному виді).

– Постачальники повинні здійснювати гарантійну підтримку поставленого встаткування й ПЗ згідно з вимогами до послуг з експлуатації й супроводу, наведеним нижче.

– Гарантійний строк на програмно-апаратне забезпечення починається з моменту приймання ПЗ або встаткування в експлуатацію й повинен тривати не менш 12 місяців для апаратного забезпечення й 6 місяців для ПЗ.

– Постачальники програмно-апаратного забезпечення повинні організувати на території України центри навчання з метою первинного навчання фахівців, що експлуатують програмно-апаратне забезпечення, їхньої перепідготовки при впровадженні нових версій і типів устаткування й ПЗ, або мати договір зі стороннім центром навчання на надання послуг з навчання, гарантуючи при цьому, що навчання на навчальних курсах даного центра буде досить для експлуатації програмно-апаратного забезпечення.

Вимоги до постачальників послуг

Загальні вимоги до постачальників послуг

До постачальників послуг пред'являються наступні вимоги:

– Перевага повинне надаватися компаніям, що надають послуги відповідно до міжнародних стандартів, прийнятими для даного типу послуг. Дана

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

відповідність рекомендується документально підтверджувати або наявністю відповідного сертифіката, або вибірковою перевіркою нормативних і робочих документів.

– Постачальник послуг повинен мати сертифікованих фахівців, як по функціоналі надаваних послуг, так і з погляду процесної організації надання послуг. Зокрема, рекомендується враховувати відповідність бізнес-процесів постачальника рекомендаціям ІТІЛ і СОВІТ по наданню послуг.

– При наданні послуг системної інтеграції рекомендується, щоб постачальник мав впроваджену в себе на виробництві систему керування проектами на основі кращих світових практик, таких як рекомендації РМІ і ІРМА. Також перевага повинне віддаватися компаніям, що мають сертифікованих керівників проектів, наприклад, по системі РМІ.

Вимоги до постачальників послугам з експлуатації й супроводу

Постачальники послуг по технічній підтримці (супроводу) і учасники конкурсів, проведених Компаніями для поставки програмно-апаратного забезпечення, повинні задовольнити наступні мінімальні вимоги до організації підтримки і експлуатації:

– Сервіс-центр повинен перебувати в прийнятній далекості від замовника, щоб мати можливість усунути несправність із виїздом до замовника в строки, установлені відповідними регламентами взаємодії, які, у свою чергу, повинні опиратися на вимоги безперервності, готовності й доступності ІТ-елементів ІТ-інфраструктури.

– Експлуатація складного програмно-апаратного забезпечення, гарантійний строк на яке минув, повинна здійснюватися на основі контракту на післягарантійне обслуговування. Зазначений контракт повинен полягати сервісом-центром, розташованим на території України й забезпечуючий післягарантійне обслуговування програмно-апаратного забезпечення даного типу. У випадках, коли програмно-апаратне забезпечення застосовується в обмежених обсягах і сервіс-центр на території України відсутній, контракт на

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

післягарантійне обслуговування полягає безпосередньо з фірмою – постачальником або її представництвом.

– Експлуатація складного програмно-апаратного забезпечення, що перебуває поза гарантійним строком, без укладеного контракту на післягарантійне обслуговування або без наявності в Замовника сервіс-центра, не допускається.

– В окремих випадках допускається здійснювати післягарантійне обслуговування силами сервіс-центра, при наявності фахівців, що пройшли відповідне навчання (що підтверджується сертифікатами постачальника), необхідного оснащення, запасних частин і оформлених належним чином взаємин із системою сервісної підтримки фірми-постачальника або виробника.

– Постачальник послуг з експлуатації й супроводу встаткування повинен мати власний склад запасних частин, що повинен перебувати в прийнятній далекості від місця установки встаткування.

– Постачальник послуг з експлуатації й супроводу повинен бути авторизований виробником на ці види діяльності. При цьому перевага повинне віддаватися постачальникам, що мають більше високий статус (рівень) відносин з виробником.

– Постачальник послуг з експлуатації й супроводу повинен мати сертифікований технічний персонал.

– Постачальники складного програмно-апаратного забезпечення повинні силами сервіс-центрів, розташованих на території України, здійснювати модернізацію програмно-апаратного забезпечення, що перебуває в експлуатації, з метою підтримки його на сучасному технічному рівні й впровадження нових сучасних функцій і послуг.

– Обслуговування й модернізація повинні здійснюватися для всіх установлених типів устаткування й ПЗ даного постачальника й для всіх версій зазначеного встаткування, у тому числі тих, поставки яких закінчилися до набрання чинності справжнім документом.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

– Пропускна здатність центрів навчання повинна бути достатньою для того, щоб забезпечувати підготовку й перепідготовку персоналу для всього діючого й знову програмно-апаратного забезпечення, що вводиться, даного типу.

Вимоги до проведення конкурсів

Учасники конкурсів, проведених Службою Замовника для поставки програмно-апаратного забезпечення й/або послуг, повинні відповідати всім додатковим вимогам, зазначеним для постачальників і виробників у даному документі, а також відповідати всім основним вимогам, які пред'являє Система стандартів по організації закупівельної діяльності.

3.2 Розробка структурної схеми

Технічні вимоги до елементів ІТ-інфраструктури

Дані технічні вимоги розглядаються в розрізі кращих світових практик по створенню ІТ-інфраструктури для сучасних корпорацій. Окремі вимоги пред'являються до наступних категорій інфраструктурних елементів:

- Робочі місця користувачів:
- Персональні комп'ютери.
- Системне ПЗ робочих місць користувачів.
- Периферійні пристрої.
- Прикладне ПЗ.

Мультисервісна мережа:

- Корпоративна розподілена мультисервісна мережа.
- Зовнішні канали зв'язку.

Інфраструктура центрів обробки даних:

- Системи обробки й зберігання даних.
- Приміщення й інженерні системи.
- Інформаційна безпека.
- Безперервність надання ІТ-послуг.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

– Системи керування й моніторингу.

Вимоги до робочих місць користувачів

Вимоги до персональних комп'ютерів

Даний розділ розглядає загальні технічні вимоги до парку персональних комп'ютерів, експлуатованих в органах державної влади, державних установах.

Загальні вимоги

При розвитку парку персональних комп'ютерів і виборі закупаваних моделей ПК ІТ-служби органів державної влади, державних установ України повинні керуватися наступними положеннями:

– Апаратна платформа й програмне забезпечення персональних комп'ютерів повинні бути стандартизовані й сертифіковані, мати гнучку й масштабовану архітектуру.

– Апаратні характеристики ПК повинні відповідати, або перевершувати мінімальні системні вимоги використовуваного ПЗ.

– Для забезпечення загального рівня послуг, керування даними всіх ПК повинне бути уніфіковане, тобто для ПК повинне бути організоване централізоване поширення програмного забезпечення за допомогою єдиного інструмента поширення відновлень програмного забезпечення.

– ПК повинен мати апаратну або програмну систему віддаленого керування.

– Для підвищення якості й швидкості адміністрування кількість різних програмно-апаратних конфігурацій персональних комп'ютерів повинне бути обмежене. Рекомендується використання не більше 4 типових конфігурацій.

Для специфікації технічних вимог виділяються наступні ключові параметри ПК:

– Продуктивність. Продуктивність персональних комп'ютерів повинна забезпечуватися за рахунок:

– Параметрів швидкодії процесора.

– Необхідного й достатнього обсягу оперативної пам'яті.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

- Швидкості внутрішніх шин передачі даних.
- Якості й швидкодії графічної підсистеми.
- Пристроїв введення/виводу.
- Надійність. Надійність повинна забезпечуватися за рахунок апаратних засобів і програмного забезпечення й визначатися виходячи із середнього часу безвідмовної роботи (MTBF).

- Масштабованість. Масштабованість повинна забезпечуватися архітектурою й конструкцією персонального комп'ютера за рахунок можливості нарощування:

- Числа й потужності процесорів.
- Обсягів оперативної й зовнішньої пам'яті.

Вимоги до типізації конфігурацій

Весь парк ПК в органах державної влади, державних установах пропонується розділити на наступні типові конфігурації:

- Персональний комп'ютер для роботи з бізнес-орієнтованим прикладним ПЗ (офісні системи, фінансові системи, СЕД і т.п.).

- Персональний комп'ютер підвищеної потужності для роботи із графічними пакетами, пакетами ПЗ моделювання, САПР, АСУЕИ, АСУТП та ін. Використовується для додатків з розвинутою графікою, високими вимогами до продуктивності процесора й обсягам оперативної пам'яті.

- Тонкий клієнт. Малопотужний персональний комп'ютер для роботи з додатками в термінальному середовищі, або із програмами-тонкими клієнтами в клієнт-серверній архітектурі. При такій роботі основні ресурсомісткі операції виробляються на сервері.

- Ноутбук для роботи мобільних користувачів. Припустимо окремо виділити ноутбук для VIP користувачів.

Вимоги до системного ПЗ робочих місць користувачів

ОС офісного призначення повинні:

- Відповідати по типі клієнтським ОС.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

– Підтримувати всі мережні сервіси, що забезпечують функціонування корпоративної мережі.

– Забезпечувати необхідний рівень інформаційної безпеки, відповідати вимогам «Політики інформаційної безпеки».

– Бути сумісними з корпоративним стандартом використовуваного офісного ПЗ.

Вимоги до периферійних пристроїв

Справжній розділ викладає основні технічні вимоги до застосовуваним і закуповуваним для служб Замовника периферійним пристроям, що входять в ІТ інфраструктуру.

Розглядаються вимоги до наступних класів периферійних пристроїв:

- Пристрою друку (принтери):
 - Монохромні принтери (персональні й групові використання).
 - Кольорові принтери (персональні й групові використання) для офісної й фотографічного друку.
 - Кольорові рулонні принтери для дизайну, картографії (персональні й групові використання).
- Пристрою сканування:
 - Сканери персональні.
 - Сканери фотографічні й дизайнерські.
 - Сканери широкоформатні.
 - Поточкові сканери.
- Багатофункціональні пристрої:
 - Персональні БФП.
 - БФП групового й корпоративного використання.
- Пристрою копіювання паперових документів:
 - Персональні копії.
 - Копії групового використання.
 - Копії корпоративні – міні друкарні.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

- Факсимільні апарати.

Вимоги до спеціалізованих пристроїв, що мають вузьке технологічне застосування (термопринтери, принтери штрих-кодів, наклейок, друкарні й т.д.), не розглядаються. Використання спеціалізованого встаткування приймається на основі конкретних технічних вимог технологічного процесу.

Для опису мінімальних вимог до периферійного встаткування використовується наступна класифікація пристроїв:

- Персональний пристрій – перебуває в постійному використанні одним співробітником.
- Груповий пристрій – використовується в режимі поділу ресурсів групою співробітників.
- Корпоративний пристрій – використовується в складі гіперЦОД I або II рівнів.

Загальні вимоги

У даному розділі викладаються загальні вимоги, які необхідно застосовувати при виборі й закупівлі нового периферійного обладнання з метою розвитку ІТ Замовника. Периферійне встаткування повинне відповідати наступним основним вимогам:

- Продуктивність. Периферійне встаткування повинне забезпечувати потреби бізнес-процесів і задовольняти вимогам, певним у кількісних показниках (наприклад, кількість копій у хвилину, дозвіл скануємого зображення й т.д.).
- Надійність. Периферійне встаткування повинне дозволяти забезпечувати безперервність бізнес-процесів і задовольняти вимогам, певним у кількісних показниках (MTBF).
- Функціональність. У тому випадку, якщо робоче місце співробітника повинне бути обладнане декількома видами периферійного встаткування, варто віддавати перевагу при закупівлі багатофункціонального пристрою (БФП), що повинне підтримувати всі або частково всі наступні функції:
 - друкованого пристрою;

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

- скануючого пристрою;
- копіювального пристрою;
- факсимільного пристрою.

– Сумісність. Периферійне встаткування повинне технічно й програмно сполучатися з персональними комп'ютерами (у випадку використання групового або корпоративного пристрою – із серверами) незалежно від типу процесора й операційної системи.

– Безпека. Вихід з ладу якого-небудь периферійного пристрою не повинен впливати на усталену роботу персональних комп'ютерів (у випадку використання групового або корпоративного пристрою – серверів) з іншим периферійним устаткуванням.

– Керованість. Підключення й керування персональним периферійним устаткуванням по можливості повинні бути простими й не вимагати оперативного використання інструкцій і описів роботи пристроїв.

– Низька ТСО. Запчастини й видаткові матеріали для периферійного встаткування повинні бути легко доступні й мати невисоку вартість.

– Низький рівень створюваного акустичного шуму. Периферійним устаткуванням в процесі роботи не повинне створювати перешкод навколишньої. Рівень шуму не повинен перевищувати 55 дБ [А]. Для експлуатації гучного встаткування повинні бути передбачені спеціально виділені приміщення.

– Низьке енергоспоживання. Рекомендується придбання встаткування, що має режим зниженого енергоспоживання (режим очікування).

– Варто віддавати перевагу тим периферійним пристроям, які в штатному режимі мають можливість обміну інформацією через локальну мережу.

– Ті периферійні пристрої, які мають можливість підключатися як до ПК, так і до локальної мережі варто підключати до локальної мережі.

Вимоги до групових і корпоративних пристроїв

Для групових і корпоративних пристроїв повинні застосовуватися більше тверді вимоги, ніж до персональних пристроїв. При виборі групових і

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

корпоративних пристроїв необхідно виходити з оцінки кількості користувачів, які будуть експлуатувати дані пристрої.

Крім зазначених вище загальних вимог, для групових і корпоративних пристроїв необхідно розглянути наявність наступного функціонала (застосовно до відповідних пристроїв):

- Наявність клієнтського додатка для ПК для прийому й розсилання факсів, а також для сканування документів.
- Підтримка формату А3.
- Потокове сканування документів з автоматичним перетворенням у потрібний формат з підтримкою мережного збереження файлів.
- Автоматична факсова розсилання.

БФП повинні проходити періодичне технічне обслуговування, що буде сприяти більшій високій доступності пристрою й знизить ТСО. Періодичність даного обслуговування необхідно визначати з технічних вимог по експлуатації кожного конкретного пристрою.

Вимоги до мультисервісної мережі

Вимоги до розподіленої мультисервісної мережі

Загальні вимоги

Основні стратегічні положення й підходи до організації корпоративної розподіленої мультисервісної мережі: орієнтація на архітектуру мереж наступного покоління (NGN).

Дана архітектура називається також архітектура softswitch (softswitch – пакетний комутатор для мереж нового покоління, є носієм інтелектуальних можливостей мережі, координує керування обслуговуванням викликів, сигналізацію й функції, що забезпечують устанавлення з'єднання через одну або кілька мереж). Дана архітектура дозволяє, міняючи версії ПЗ на Softswitch-ах і самі Softswitch-і, розширювати функціональні можливості, не міняючи встаткування доступу. Крім цього, відповідно до архітектури IMS (Interactive Multimedia Subsystem), стандартизується протокол надання сервісів, що дозволяє,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

без доробки інтерфейсів взаємодії, підключати будь-які послуги від будь-яких сервіс-провайдерів.

Телефонія, аудіоконференцзв'язок, відеоконференцзв'язок і передача даних повинні ґрунтуватися на єдиній конвергентній мультисервісній мережі, що дозволяє надавати зазначені сервіси, забезпечуючи необхідний і достатній рівень QoS.

NGN-мережа побудована на наступних принципах:

- Розподілена корпоративна архітектура, що забезпечує QoS.
- Висока доступність і надійність мережі.
- Продуктивність, керованість і масштабованість мережі.
- Мультисервісна мережа повинна проектуватися з урахуванням вимог інформаційної безпеки.

Весь трафік повинен бути класифікований на наступні класи, що визначають пріоритет обслуговування:

- Трафік реального часу (телефонний, аудіо й відео).
- Трафік передачі користувальницьких даних.
- Технологічний трафік.

Якщо технологічне встаткування вимагає найвищого класу обслуговування, то його трафіку повинен бути відданий найвищий пріоритет. Після цього повинен іти трафік реального часу, після якого треба трафік користувальницьких додатків. Причому пропускна здатність мережі й активне мережне встаткування повинні завжди забезпечувати якість для трафіку реального часу.

При аварійних ситуаціях ресурси мережі повинні віддаватися технологічному трафіку й частини трафіку реального часу й користувальницького трафіку в тому обсязі, у якому це необхідно для забезпечення безперебійної роботи основного технологічного встаткування й обслуговуючого його персоналу. Даного правила повинні бути реалізовані в

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

настроюваннях устаткування й набувати чинності автоматично при настанні аварійної ситуації.

Вимоги до архітектури мережі

Архітектура мультисервісної мережі повинна бути заснована на наступних принципах:

- Будуватися на трирівневої моделі.
- Мати резервування по встаткуванню й каналам.
- Підтримувати VLAN.
- Архітектурно розбиватися на демілітаризовані зони (ДМЗ).
- Мережа повинна забезпечувати необхідну для рішення завдань продуктивність.

- Мережа повинна забезпечувати QoS для різних класів трафіку.

Мультисервісна мережа для всіх гіперЦОД повинна проектуватися, виходячи із трирівневої моделі комутації:

– Рівень доступу (Access Layer) – комутатори 2-го рівні моделі OSI з інтелектуальністю 3– 4-го рівнів моделі OSI (з метою забезпечення вимог до мережної безпеки, QoS і т.д.).

– Рівень розподілу (Distribution Layer) – комутатори 3– 4-го рівнів моделі OSI.

– Магістральний рівень / рівень ядра (Core Layer) – комутатори 3- 4-го рівнів моделі OSI.

У випадках, коли використання виділених комутаторів рівня розподілу в якому-небудь сегменті мережі недоцільно через зниження продуктивності мережної інфраструктури, зниження надійності або в силу інших причин, допускається переносити функції комутаторів рівня розподілу на комутатори рівня ядра.

Структурна схема зазначеної трирівневої моделі гіперЦОД наведена на рисунку 3.1.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

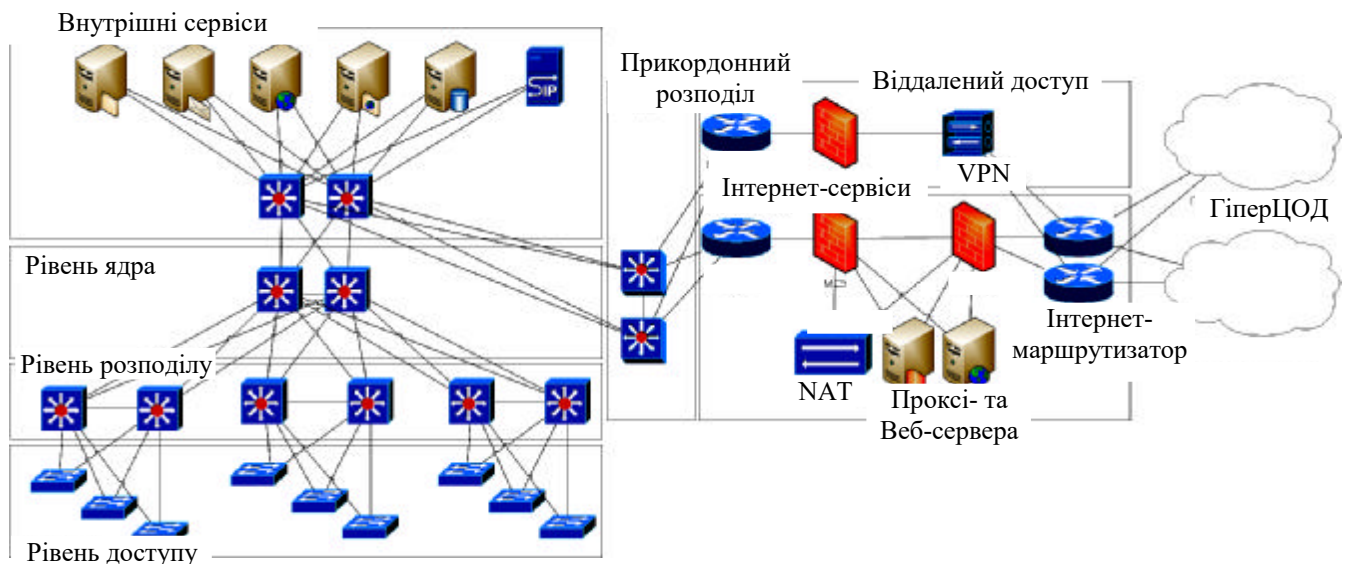


Рисунок 3.1 – Структурна схема системи

Обрана архітектура мережі повинна дозволяти нарощувати мережу шляхом додавання нових блоків, забезпечувати високий детермінізм поведінки мережі, вимагати мінімальних зусиль і засобів для пошуку й усунення несправностей. Інтелектуальні сервіси 3-го рівні моделі OSI (у тому числі протоколи маршрутації) повинні забезпечувати скорочення області, що зачіпається при виникненні різноманітних проблем з несправним або невірно настроєним устаткуванням, а також балансування навантаження між/усередині рівнями ієрархії й швидку збіжність.

Повинні бути дотримані загальні правила проектування трирівневої структури:

- Будь-які проблеми з устаткуванням і каналами зв'язку на нижчележачих рівнях не повинні позначатися на верхніх рівнях.
- Транзитні резервні маршрути певного рівня не повинні проходити через нижчележачі рівні.
- Класифікація трафіку повинні відбуватися тільки на рівні доступу. Пріоритизація трафіку повинна підтримуватися всіма рівнями мережі. Рівень

розподілу повинен тільки агрегувати трафік. Ядро мережі повинне тільки здійснювати швидку комутацію й маршрутизацію пакетів.

– Час збіжності таблиць маршрутизації і їхній обсяг повинні бути оптимізовані для кожного рівня за допомогою вибору оптимальної схеми резервування.

– Віддалені користувачі й зовнішні канали зв'язку не повинні приєднуватися прямо до ядра мережі. Необхідно використовувати комутатори доступу для запобігання лавиноподібних налаштувань таблиць маршрутизації всієї мережі.

– Забороняється використання некерованих комутаторів. Мінімально припустимий комутатор у мережі – керований комутатор рівня 2.

Резервування для гіперЦОД I рівня повинне, а для гіперЦОД II рівня рекомендується організувати в такий спосіб:

– У мережі повинне бути не менш двох комутаторів рівня ядра, зв'язаних між собою по 10 (або вище) GigabitEthernet, або об'єднаних в відказостійкий стек з еквівалентною пропускною здатністю.

– Кожний комутатор рівня доступу повинен мати з'єднання каналами Gigabit Ethernet із двома комутаторами рівня розподілу.

– Кожний комутатор рівня розподілу повинен мати з'єднання каналами Gigabit Ethernet (або вище) із двома комутаторами рівня ядра.

– Для забезпечення відказостійкості в мережі повинне бути два прикордонних маршрутизатори. Маршрутизатори підключаються кожний до не менш чим двох різних інтернет-провайдерів і здійснюють маршрутизацію пакетів по протоколу BGP. Кожний прикордонний маршрутизатор повинен бути пов'язаний із двома пристроями, що забезпечують функціональність міжмережного екрана (MME) або IDS/IPS.

– Для забезпечення незалежності від інтернет-провайдерів повинна використовуватися автономна системи (AS) із власним пулом ір-адрес (не менш /22).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

схемою 1:N (при виході з ладу одного з комутаторів стека, незалежно від виконуваної їм функції, інші будуть продовжувати виконання своїх функцій без зупинки роботи всієї мережі).

– Рекомендується використовувати динамічний протокол внутрішньої маршрутизації OSPF як володіє гарною масштабованістю, швидкою збіжністю, що враховує якість каналів зв'язку й займає мінімальну смугу каналу.

Система IP-адресації мережі повинна забезпечувати:

– Поділ адресного простору на службовий блок (мережі, що зв'язують маршрутизатори, віртуальні інтерфейси й т.п.) і блок адрес локальної мережі. Такий поділ дозволяє ефективно будувати правила доступу до мережних пристроїв.

– Розподіл адресного простору локальної мережі блоками відповідно до територіального розташування. Такий поділ дозволяє робити агрегування адрес, що приводить до зменшення таблиць маршрутизації й спрощує керування мережею.

Для здійснення автентифікації на рівні доступу для всіх пристроїв, що забезпечують функціонування мережі, і при доступі до консолі керування всіма пристроями, мережа повинна мати наступні можливості:

– Безпека портів, тобто повинна бути можливість використання порту комутатора з попередньо заданими фізичними адресами користувальницьких ПК (MAC-адреси). При спробі підключення неавторизованого пристрою повинне вироблятися відключення цього порту й повідомлення системи керування мережею.

– Автоматичне конфігурування портів комутаторів, тобто повинна бути автоматизація зміни конфігурації порту на основі логічного підключення користувача до мережі.

– Автентифікація адміністративного доступу на Radius сервері, тобто повинна бути ідентифікація, авторизація й облік при доступі до командного рядка пристрою.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

– Обмеження доступу по IP адресах, з урахуванням обмеження на доступ до командного рядка пристрою й системної консолі, а також SNMP трафіку.

– Повинна бути автоматична фільтрація трафіку невикористовуваних протоколів на портах комутаторів.

Для забезпечення високої доступності мережі рекомендується використовувати наступну функціональність:

– Підтримка протоколу RSTP/MSTP або інших протоколів резервування другого рівня.

– Підтримка можливості поєднувати в єдиний логічний канал кілька фізичних з'єднань між комутаторами.

– Функції автоматичного перемикання з основного маршрутизатора на резервний у випадку відмови основного.

– Балансування навантаження між резервуваними маршрутизаторами.

– Функції внутрішнього програмного забезпечення для поліпшення часу збіжності протоколів маршрутизації й балансування навантаження через рівноцінні маршрути.

Для підтримки додатків, заснованих на технології багатоадресного розсилання IP Multicast, від мереж потрібна наявність наступних можливостей:

– На рівні доступу/розподілу – передача пакетів IP Multicast на каналному рівні на швидкості фізичного каналу, динамічна реєстрація за допомогою протоколів IGMP і PIM.

– Магістральний рівень – передача пакетів IP Multicast на каналному й мережному рівнях на швидкості фізичного каналу, масштабовані протоколи маршрутизації трафіку IP Multicast.

Вимоги до телефонії, аудіо- і відео-конференцзв'язку

Основний протокол передачі аудіо- і відеоінформації – IP. Допускається використання традиційної аналогової телефонії в наступних випадках:

– Успадковані існуючі телефонні станції.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

– Економічна доцільність. Дане виключення діє до моменту, коли вартість VoIP телефонів стане порівнянна із ціною аналогового телефону.

VoIP переважно повинна впроваджуватися за технологією SIP, тому що дана технологія, у порівнянні з Н.323, використовується в мережах наступного покоління й має більшу функціональність. При цьому необхідно звертати увагу на сумісність реалізації протоколу SIP між прикінцевими пристроями й програмно-апаратним комплексом, що забезпечує функціональність телефонної станції. Дана сумісність повинна виражатися в підтримці основного функціонала по обробці вступників дзвінків з прикінцевих пристроїв. З метою недопущення проблем, пов'язаних з несумісністю реалізації протоколу SIP, рекомендується встановлювати прикінцеві пристрою (телефони) і програмно-апаратні комплекси, що забезпечують функціональність телефонної станції, одного виробника, або проводити ретельне лабораторне тестування сумісності апаратури різних виробників.

Системи відеоконференцзв'язку повинні підтримувати Web-конференції й інтегруватися з офісними додатками.

Сервера аудіоконференцзв'язку повинні підтримувати або мати можливість розширення для підтримки відеоконференцзв'язку.

Відеоконференцзв'язок повинен організовуватися на технології IP з використанням стандартів Н.323/Н.264.

При пакетної передачі за еталон якості мови повинен бути прийнятий рівень якості, рівний 4 балам по шкалі MOS/PAMS (Mean Opinion Score, суб'єктивний метод оцінки відповідно до рекомендації Р.800). Рекомендується використовувати кодек G.729 (MOS = 4.07). Вимоги до параметрів якості пакетної передачі: затримка пакетів – до 150 мс, джиттер – до 50 мс.

При наявності двох і більше провайдерів, включаючи традиційну телефонію й VoIP, рекомендується використовувати LCR – вибір провайдеру по найменшій вартості. При цьому встаткування VoIP повинне забезпечувати моніторинг якості каналу зв'язку.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Вимоги до встаткування

Устаткування рівня доступу повинне мати можливість класифікації трафіку (Traffic Classification), тобто повинна бути забезпечена можливість класифікувати трафік по типах додатків, фізичним і мережним адресам джерел і одержувачів, портам комутаторів. Класифікований трафік повинен одержувати мітку, що позначає призначений пакетам рівень пріоритету, тим самим даючи можливість пристроям мережі відповідним чином обслуговувати цей трафік. Повинна бути забезпечена рекласифікація пакетів на основі заданій адміністратором політики якості обслуговування. Наприклад, користувач призначає високий пріоритет своєму трафіку й передає його в мережу. Цей пріоритет може потім бути знижений відповідно до мережної політики, а не на основі вимог користувача. Даний механізм повинен бути ключовим у забезпеченні якості обслуговування в рамках всієї мережі.

Устаткування магістрального рівня повинне мати наступну функціональність:

– Запобігання й керування перевантаженнями, тобто повинна бути забезпечена можливість управляти поведінням мережі при перевантаженнях, відкидаючи певні пакети на основі класифікації або політики в моменти перевантаження мережі й безлічі черг на інтерфейсах. Адміністратор повинен установлювати граничні значення для різних рівнів пріоритету.

– Планування, тобто повинна бути забезпечена можливість здійснювати пріоритетну передачу пакетів, засновану на класифікації або політику якості обслуговування, за допомогою декількох черг.

– Резервування основних вузлів, до яких може ставитися: блок живлення, блок вентиляторів, процесорний модуль.

– Надавати можливість поглибленого аналізу потоків мережного й транспортного рівнів за допомогою протоколу IPFIX (RFC 3917), Netflow, J-Flow або іншого протоколу надання агрегованої статистики по ір-потоках.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

У гіперЦОД I і II рівня крім забезпечення резервування основних вузлів устаткування магістрального рівня, рекомендується забезпечити таке ж резервування для встаткування рівня розподілу.

У всьому активному мережному встаткуванні повинні бути засоби моніторингу політики якості обслуговування й безпеки, планування мережі й сервісів:

– Повинна бути забезпечена можливість збору статистики з точністю до порту мережі для аналізу продуктивності й виявлення вузьких місць мережі.

– Повинна бути забезпечена можливість перенаправляти трафік окремих портів, груп портів і віртуальних портів на аналізатор протоколів для детального аналізу.

– Повинна бути забезпечена можливість розширеного моніторингу подій у реальному часі для розширення можливостей діагностики, крім зовнішніх аналізаторів.

– Повинна бути забезпечена можливість збору й збереження інформації про істотні мережні події, включаючи зміни конфігурацій пристроїв, зміни топології, програмні й апаратні помилки за технологією syslog.

– Повинна бути забезпечена можливість доступу до інтерфейсу керування пристроєм і звітам через стандартний WEB-Браузер з використанням протоколу HTTPS.

– Повинна бути можливість підключення до пристрою для його налаштування з використанням протоколу SSH.

– Повинна бути забезпечена можливість автоматичної конфігурації Fast/Gigabit Ethernet портів, віртуальних мереж, транків VLAN.

– Повинна бути забезпечена можливість автоматичного розпізнавання топології мережі за допомогою агентів розпізнавання топології.

– З метою забезпечення продуктивності локальної мережі, її масштабованості, задоволення вимогам інформаційної безпеки й забезпечення якості обслуговування мультисервісного трафіку, забороняється використовувати

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

концентратори (hub). Замість них повинні використовувати тільки комутатори (switch).

Все активне встаткування повинне мати конструктивне 19” стоечне виконання.

Вимоги до зовнішніх каналів зв'язку

Даний розділ розглядає технічні вимоги до зовнішніх каналів зв'язку, які надаються сторонніми операторами зв'язку.

Загальні положення

З метою уніфікації необхідно виділити наступні використовувані види каналів зв'язку:

- Телефонні цифрові потоки E1 PRI.
- Виділені канали передачі даних із шириною від 64 кбіт/с до 2 Мбіт/с і вище.
- Орендовані канали мережі передачі даних.

При виборі виду каналу зв'язку перевага необхідно віддавати каналам зв'язку, які підключаються до мережі MPLS оператора, тому що тільки мережі MPLS у цей час ефективно забезпечують QoS для мультисервісного трафіку при прийнятній вартості послуги. Інтерфейс підключення, що рекомендується – Ethernet, точка-точка.

Канали зв'язку для з'єднань точка-точка або точка-багатоточка між гіперЦОД I і II рівнів повинні організовуватися за допомогою технології MPLS або іншої технології, що забезпечує виконання необхідних вимог по пропускній здатності каналу і якості надання послуги, що підтримує оператор зв'язку. При цьому повинен бути укладений договір, що передбачає забезпечення QoS для аудіо- і відеоданих, якщо такі є. Вимоги повинні бути зазначені у відповідному SLA.

При підключенні гіперЦОД I і II рівнів оператор зв'язку повинен забезпечити цілодобову службу технічної підтримки, що у будь-який час доби не тільки приймає заявки, але й усуває інциденти.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

3.3 Розробка функціональної схеми

Прикладне програмне забезпечення (ПЗ)

Прикладне програмне забезпечення (прикладне ПЗ) є однією з основних компонентів сучасної ІТ-інфраструктури. З погляду кінцевого користувача, саме прикладне ПЗ допомагає вирішувати ті або інші бізнес-завдання.

У рамках даного документа ми будемо розглядати наступні основні характеристики прикладного ПЗ:

- Функціональність – здатність ПЗ максимально ефективно виконувати заявлені функції, з необхідними характеристиками.
- Функціональна повнота – повний набір функцій, які здатно виконувати дане ПЗ.
- Платформонезалежність – здатність ПЗ функціонувати в різних програмно-апаратних середовищах, під керуванням різних операційних систем.
- Продуктивність – здатність ПЗ забезпечувати збір, обробку й зберігання певних обсягів інформації при заданій конфігурації апаратної платформи.
- Масштабованість – здатність ПЗ коректно працювати на малі й на більших системах із продуктивністю, що збільшується пропорційно обчислювальної потужності системи (кількості процесорів, дискових масивів, серверів і т.д.), використовуваних для експлуатації даного ПЗ.
- Здатність до інтеграції (Open System Interconnection (OSI), Interoperability) – здатність ПЗ до інтеграції й взаємодії з іншими системами, у тому числі із системами сторонніх виробників, і експлуатованими на інших платформах.
- Зрілість – наявність історії розвитку даного ПЗ (присутність даного ПЗ на ринку протягом певного проміжку часу, регулярний випуск виробником нових версій і релізів) і оголошених виробником планів по його розвитку.
- Надійність і відказостійкість – здатність ПЗ й систем, побудованих на його базі, до безперебійної, безперервної роботи, а у випадку виникнення

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

програмно-апаратних збоїв – здатність до швидкого відновлення працездатності системи.

– Загальна вартість володіння (ТСО) ПЗ – складається з витрат на початкове придбання (розробку) даного ПЗ, введення його в експлуатацію й витрат на його експлуатацію й супровід протягом нормативного строку життя даного ПЗ. Загальна вартість володіння включає витрати: на відновлення програмного забезпечення й устаткування. на навчання, обслуговування, адміністрування й технічну підтримку.

З погляду процесів розробки, поставки й супроводи всю сукупність прикладного ПЗ можна розділити на дві групи:

– Універсальне (тиражуєме) ПЗ – ПЗ, доступне на ринку й, що служить для рішення універсальних завдань.

– Замовлене ПЗ – ПЗ, розроблене за замовленням ІТ-підрозділом або сторонньою організацією.

При наявності на ринку ПЗ з відкритим кодом з функціональністю, надійністю й зручністю використання порівнянними або переважаючими за аналогічними показниками ПЗ із закритим кодом перевага у використанні повинне віддаватися ПЗ з відкритим кодом.

Загальні вимоги до прикладного ПЗ

При виборі й впровадженні нового прикладного ПЗ повинні дотримуватися наступні вимоги:

– Все використовуване прикладне ПЗ повинне бути уніфіковане й каталогізовано в рамках КРК у вигляді списку ПЗ, припустимого до використання.

– ПЗ клієнтських ПК повинне бути функціонально повним і забезпечувати виконання як стандартних бізнес-процесів, так і специфічних бізнес-завдань даного користувача.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

– ПЗ повинне бути зрілим: виробник (постачальник) повинен гарантувати підтримку, супровід даного ПЗ протягом усього нормативного строку життя даного ПЗ.

– Рекомендується використовувати платформонезалежне ПЗ, що забезпечує волю вибору програмно-апаратних засобів для його експлуатації й, в остаточному підсумку, зниження загальної вартості володіння системою.

– Рекомендується використовувати продуктивне, масштабоване ПЗ, що забезпечує гарантії безперервності бізнес-процесів при росту обсягів оброблюваної й збереженої інформації. Бажано, щоб виробник ПЗ регулярно проводив об'ємне й навантажувальне тестування свого ПЗ й надавав дані про результати даного тестування.

– Рекомендується використовувати тільки ПЗ, що володіє здатністю до інтеграції з іншими системами, і володіюче відкритою архітектурою, тобто підтримуючий стандарти OSI. При виборі ПЗ необхідно враховувати можливості його інтеграції в існуючу ІТ інфраструктуру підприємства.

– При виборі ПЗ переваги повинні одержувати системи з підтвердженим позитивним досвідом використання в державних установах.

– Для забезпечення критичних бізнес процесів і послуг рекомендується використовувати відказостійке ПЗ.

– Загальна вартість володіння ПЗ, розрахована на весь нормативний строк експлуатації даного ПЗ, повинна служити найважливішим критерієм при виборі того або іншого постачальника й ПЗ.

– В області інформаційної безпеки ПЗ повинне відповідати вимогам політики Уряду в області інформаційної безпеки.

– ПЗ повинне бути належним чином документовано. Мінімальні вимоги до документації – наявність документів «Посібника користувача» і «Керівництво адміністратора».

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Загальні вимоги до універсального прикладного ПЗ

При закупівлі нового універсального прикладного ПЗ повинні дотримуватися наступні вимоги:

- Рекомендується закуповувати ПЗ в рамках спеціальних моделей ліцензування, що забезпечують зниження вартості закупівлі.
- При використанні ліцензуємого ПЗ на нього повинні бути обов'язково отримані й належним чином зареєстровані відповідні ліцензії.
- Рекомендовано до використання ПЗ виробників, що зарекомендували себе на ринку в даній області. Бажано, щоб даний виробник був присутній на ринку з даним або аналогічним ПЗ не менш трьох років. Не рекомендується використання застарілих версій ПЗ, а також занадто нових, «незрілих» версій ПЗ.
- Повинне бути заборонене до використання ПЗ, що не має як ліцензій, так і підтримки (супроводу) з боку виробника.

Загальні вимоги до замовленого прикладного ПЗ

При розробці нового прикладного ПЗ повинні дотримуватися наступні вимоги:

- Процес проектування, розробки й впровадження замовленого ПЗ повинен відповідати вимогам розділу 6.6. «Вимоги до створення систем і запровадженню в дію. Вимоги до документації» справжнього документа.
- необхідності техніко-економічного обґрунтування розробки й впровадження даного ПЗ.
- При формуванні Технічного завдання відповідно до ДСТУ 34.602-89 у розділі «Вимоги до системи» необхідно докладно сформулювати й описати вимоги до замовляємого ПЗ, зокрема, до його основних властивостей, перерахованим вище. Замовник ПЗ повинен представити техніко-економічного обґрунтування розробки й впровадження даного ПЗ, з обліком бізнес-вимог і принципів мінімізації ТСО для даного рішення.
- При виборі розроблювача ПЗ основну увагу повинне приділятися досвіду попередньої роботи даного розроблювача по створенню (проектуванню,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

реалізації й впровадженню) подібних систем. Бажана вимога реалізації не менш трьох аналогічних по функціоналі й функціональній повноті проектів протягом останніх двох років.

– При виборі розроблювача ПЗ необхідно сформулювати вимоги до застосовуваних систем керування якістю. Бажана наявність у розроблювача сертифіката на відповідність його системи керування якістю процесів розробки ПЗ стандартам сімейства ISO 9000.

– Рекомендується включати в процедури приймання ПЗ передачу розроблювачем вихідних текстів програм і інших об'єктів, необхідних для створення ПЗ, відповідно до вимог ДСТУ ISO/МЕК 12.207-99. Процедура приймання повинна обов'язково містити в собі контрольну компіляцію переданих вихідних текстів, зі створенням повністю працездатної версії ПЗ, і виконання контрольного приклада на даній версії.

– У договорі на розробку ПЗ необхідно відбивати розподіл авторських і суміжних прав на кінцевий продукт, а також обмеження на його подальше використання сторонами.

Вимоги до інфраструктури гіпер центрів обробки даних (гіперЦОД)

Вимоги до систем обробки й зберігання даних

Даний розділ розглядає технічні вимоги до систем зберігання й резервного копіювання даних.

Загальні вимоги

Загальні вимоги до систем обробки, зберігання й резервного копіювання даних для всіх гіперЦОД:

– Продуктивність. Продуктивність устаткування повинна складатися із продуктивності основних підсистем. Необхідно відслідковувати навантаження основних підсистем, виявляти вузькі місця й нарощувати, у міру необхідності, продуктивність шляхом оптимізації конфігурації, установки додаткових модулів або заміни поточних модулів на більше продуктивні. У робочому режимі сервер

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

повинен мати завантаження основних ресурсів не більше ніж на 75% , щоб витримувати пікове навантаження якщо буде потреба.

– Віртуалізація. Рекомендується використовувати встаткування, що підтримують віртуалізації як при обробки інформації (підтримка віртуалізації на апаратному рівні використовуваних серверів), так і при зберіганні (віртуальні диски на системі зберігання даних) і резервному копіюванні інформації (віртуальні стрічки, використання технологій D2D, або D2D2T).

– Масштабованість. Необхідно використовувати встаткування, що має, якщо буде потреба, можливість нарощування Готовність. Ступінь готовності встаткування повинна забезпечуватися за рахунок:

- зменшення одиничних точок відмови.
- технології об'єднання декількох серверів у кластер.
- використання систем високої готовності від провідних виробників.

Сервери

Вибір серверного встаткування повинен залежати від тих завдань (додатків), які вони будуть вирішувати. З огляду на, що розкид завдань величезний і, при зростанні кількості користувачів і обсягів даних, вимоги до обчислювальних ресурсів різко підвищуються, то рекомендується:

– Вибирати сервери, що дозволяють поступово масштабувати ресурси й збільшувати продуктивність.

– Використовувати технологію віртуалізації, що дозволяє розділяти ресурси високопродуктивного сервера між додатками які вимагають не дуже більших ресурсів для своєї реалізації, на апаратному й програмному рівнях.

Даних підхід, що сполучить у собі установку масштабованих серверів і технологію віртуалізації, дозволить зменшити ТСО і збільшити прозорість і керованість всією обчислювальною інфраструктурою за рахунок динамічного перерозподілу ресурсів.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Сервери повинні забезпечувати:

- Високу швидкість обробки даних при знижених витратах на обслуговування.
- Простоту керування для швидкої зміни й перерозподілу ресурсів залежно від потреб.
- Високу надійність і безперервність обробки й доступу до інформації.
- Інтеграцію їх в існуючу інфраструктуру й спільну роботу із уже, що використовуються системами, обробки даних.
- Бути енергоефективними.

Рекомендується встановлювати в гіперЦОД I і II рівнів високопродуктивні серверні платформи провідних виробників середнього й вищого рівнів продуктивності, у гіперЦОД III рівня – початкового й середнього рівнів.

При виборі серверів рекомендується віддавати перевагу платформам, що підтримують не менш двох процесорів із двоядерною архітектурою.

Високопродуктивні серверні платформи повинні мати ряд убудованих систем високої доступності, таких як: резервні вентилятори й блоки живлення гарячої заміни. диски й адаптери I/O гарячого підключення. динамічне очищення й перерозподіл сторінок пам'яті. динамічний перерозподіл процесорів і здатність до відновлення. інтегрована служба оповіщення про події, що працює в режимі реального часу. убудована розширена система виявлення несправностей з виділеним сервісним процесором і шиною. наявність віддаленої консолі.

У всіх серверних рішеннях повинне бути приділене велика увага запобіганню можливих збоїв. Для гіперЦОД I і II рівнів повинні бути реалізовані відповідні функції, за допомогою яких здійснюється безперервний контроль стану всіх компонентів сервера й аналіз тенденцій зміни контрольованих показників. При виявленні якої-небудь потенційної проблеми, наприклад можливого перегріву процесора, спеціальні функції динамічного перерозподілу ресурсів повинні забезпечити перенос процесів з потенційно-збійного компонента на справний без переривання виконання додатків. При цьому

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

адміністратор системи й/або служба технічної підтримки повинні одержати повідомлення й докладний звіт про подію, що відбулася.

ПЗ, що реалізує технологію віртуалізації серверів, повинне давати можливість швидко й просто розділяти обчислювальні ресурси залежно від вимог додатків, а також зменшувати загальне число серверів, дозволяючи декільком віртуальним серверам розміщатися на один фізичному, раціонально використовуючи його обчислювальні ресурси й пам'ять.

ПЗ, що реалізує технологію віртуалізації серверів, повинне реалізовувати наступну функціональність:

Функція декомпозиції:

- комп'ютерні ресурси повинні розглядатися як єдиний однорідний пул, що розподіляється між віртуальними машинами;
- безліч додатків і операційних систем повинні співіснувати на одній фізичній комп'ютерній системі.

Функція ізоляції:

- віртуальні машини повинні бути повністю ізольовані друг від друга.

Аварійна відмова однієї з них не повинен робити ніякого впливу на інші;

- дані не повинні передаватися між віртуальними машинами й додатками, за винятком випадку використання загальних мережних з'єднань зі стандартною конфігурацією.

Сумісність:

- Сумісність повинна гарантуватися за допомогою подання віртуальної апаратури додаткам і ОС як стандартної.

У гіперЦОД I рівня обов'язково, а в гіперЦОД II рівня рекомендується використовувати дисковий масив для зберігання основних прикладних даних і функцію зберігання даних передавати мережі зберігання даних (SAN) з наступної віртуалізацією SAN. Убудовані диски серверів використовувати тільки для системних цілей, використовуючи технологію надмірності RAID.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Якщо для гіперЦОД II рівня не використовуються мережі SAN, то повинні бути використані пристрої NAS з функціональністю віртуалізації.

Таким чином, для гіперЦОД I і II рівнів повинна бути впроваджена технологія віртуалізації на всіх рівнях IT-інфраструктури:

- Локальна мережа – технологія VLAN.
- Сервера – технологія віртуалізації серверів.
- Мережі й системи зберігання даних – технологія віртуалізації SAN або NAS.

ОС для обслуговування серверних додатків і промислових систем повинні:

- Бути високонадійними й захищеними.
- Забезпечувати високий рівень швидкодії додатків.
- Мати убудовані можливості для організації віддаленого моніторингу всіх основних сервісів ОС.

Мережі й системи зберігання даних, електронні архіви

Головними пріоритетами в розвитку систем зберігання даних повинні бути:

- нарощування ємності систем зберігання даних;
- концентрування систем зберігання даних у єдиному місці, причому кількість територіально-розподілених місць повинне бути обмежене;
- розширення можливостей відновлення після аварій;
- зменшення часу відновлення;
- зменшення вікон резервного копіювання (інтервалів часу, відведених для підготовки резервної копії) для критично важливих додатків.

Повинні бути виділені наступні рівні системи зберігання даних:

- Зверхоперативний рівень.
- Оперативний рівень.
- Рівень довгострокового зберігання даних.
- Електронний архів.
- Резервного копіювання даних.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

передачі даних як технологію FibreChannel, так і технологію Ethernet з використанням протоколу iSCSI.

Всі ці рішення поєднує одна характеристика – спроба знизити ТСО системи зберігання, впроваджуючи ефективне керування централізованою інформацією, ізольовано розташованими в гетерогенному середовищі, що включає різні ОС, формати даних і користувальницькі інтерфейси.

У гіперЦОД I рівня повинна бути впроваджена технологія SAN, як задовольняючим всім необхідним вимогам. Для гіперЦОД II рівня також рекомендується технологія SAN, але допускається можливість розгортання пристроїв NAS для організації файлових сервісів. При цьому повинна бути стратегічно обрані одна технологія або продумана й обґрунтований підхід одночасного використання й SAN і NAS без взаємних конфліктів, що забезпечує віртуалізацію для серверних систем і загальне керування єдиним пулом гіперЦОД.

Метою впровадження й використання технології SAN повинне стати забезпечення реальної консолідації ресурсів зберігання і їхнього спільного використання, тому що ємність зберігання повинна підключатися до багатьом серверам, у тому числі й віддаленим, а машини, що обробляють дані, повинні звільнитися від завдань керування ресурсами і їхнім зберіганням.

При впровадженні технології SAN повинне бути забезпечене:

- незалежність топології SAN від storage-систем і серверів;
- зручне централізоване керування;
- зручне резервування даних без перевантаження локальної мережі й серверів;
- висока швидкодія;
- висока масштабованість;
- висока гнучкість;
- висока готовність.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

При виборі й впровадженні конкретного встаткування й ПЗ для реалізації SAN необхідно дотримувати наступної вимоги: система зберігання даних повинна підтримувати рівні логічної абстракції (віртуалізації) між фізичними портами на даному дисковому масиві, блоками даних на конкретних дискових групах і логічних томах або файлами, до яких сервери або додатки повинні мати доступ.

Зокрема повинні бути реалізовані наступні сервіси віртуалізації:

– Віртуалізація підключення до SAN. До дискового масиву через SAN повинні одержувати доступ кілька серверів і розподіл фізичних портів масиву між ними не повинне бути управлінською проблемою й не повинне стати перешкодою для повного використання можливостей гіперЦОД. ГіперЦОД повинна дозволяти створювати кілька віртуальних портів на одному фізичному порту Fibre Channel, а також управляти цими портами.

Віртуалізація логічних дисків і томів:

– Будь-яка модифікація додатка (наприклад, додавання нових серверів, пристроїв зберігання даних або функцій) вимагає виконання складного комплексу дій по зміні настроювань, як на серверах, так і на дискових масивах. Ці дії не повинні бути причиною помилок і простоїв, і не повинні збільшувати час, необхідне на розгортання й модифікацію додатка.

– ГіперЦОД повинна забезпечувати надійні сервіси керування логічними дисками (LUN) і томами для поширення розширених сервісів керування інформацією й зберіганням даних на модульні системи зберігання даних, що підтримують різні типи дисків.

– Необхідно мати відповідне апаратне забезпечення із продуктивністю, достатньої для значного підвищення масштабованості й гнучкості рішень по віртуалізації без шкоди для доступності даних або без збільшення витрат на керування системою зберігання даними.

При побудові SAN у гіперЦОД I рівня на комутаторах рекомендується створювати дві незалежні Fabric (Dual fabric). Dual fabric дозволяє уникнути

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

єдиної точки відмови в SAN, забезпечуючи високий рівень надійності й відказостійкості. Крім того, зміни конфігурації, регламентні роботи (наприклад, установка нового firmware) на комутаторах однієї з Fabric не позначаються на роботі іншої. Застосування Dual fabric разом із програмним забезпеченням, що реалізує підтримку альтернативних шляхів доступу й розподіл навантаження, для з'єднання серверів і пристроїв зберігання (шляхи повинні бути розподілені між різними Fabric) дозволяє створити надійну SAN. Також необхідно передбачати наявність на серверах ПЗ Dynamic multipathing для забезпечення безперервної роботи додатків із двома фабриками.

Рекомендується вибирати встаткування, що підтримує Fibre Channel із пропускнуою здатністю 8 Гбит/із із підтримкою швидкостей передачі даних 4/2/1 Гбит/с. У випадку забезпечення більше високої швидкості передачі даних по магістральних каналах необхідне використання транкінга (trunking) – об'єднання декількох каналів передачі даних в один канал.

Забезпечення високої доступності додатків

У тих випадках, коли потрібно забезпечити високу надійність і доступність додатків і інформаційних систем для користувачів, необхідно використовувати технології кластеризації додатків. Залежно від необхідної величини надійності й доступності системи, яку необхідно забезпечити, доцільно застосовувати або кластери, що працюють у режимі активний/резервний (high-availability clusters), або паралельні кластери (parallel clusters), що забезпечують більше високий рівень доступності. При виборі прикладного ПЗ необхідно враховувати можливості систем по роботі в кластерних конфігураціях.

Резервне копіювання даних

Резервне копіювання даних ІС для всіх гіперЦОД повинне здійснюватися відповідно до "Політики резервного копіювання", що повинна містити загальний опис процесів резервного копіювання. Доповнення до "Політики резервного копіювання" містять процеси резервного копіювання для різних ІС і регламенти їхнього виконання.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Забезпечення катастрофостійкості

Катастрофостійкість – здатність комп'ютерного комплексу, що складає з декількох систем, зберегти критично важливі дані й продовжити виконувати свої функції після масового (можливо, цілеспрямованого) знищення його компонентів у результаті різних катаклізмів як природного характеру, так і інспірованих людиною.

Катастрофостійкість припускає в першу чергу забезпечення схоронності даних, а також можливість відновлення роботи після великої локальної аварії або глобального катаклізму, причому тими ж засобами забезпечується заодно й належний ступінь надійності всіх або критично важливих підсистем. Оскільки компоненти розподілені, то у випадку масових відмов на одній площадці основну роботу можна перенести на іншу площадку.

Збереження даних можна забезпечити як засобами гіперЦОД (синхронне/асинхронне дзеркалюванні даних на резервний гіперЦОД), так і засобами додатків. Наприклад, при роботі із СУБД налаштовується пересилання у віддалений гіперЦОД журналів змін, які ведуть більшість СУБД. Резервний центр при цьому не обслуговує користувачів, однак зобов'язаний мати комплект устаткування й ПЗ для підтримки відповідної бази даних потрібного обсягу, здатний устигати вносити в неї зміни в міру надходження журналів з основного центра.

Також для забезпечення катастрофостійкості необхідно на резервному ЦОДі мати обчислювальні потужності не нижче використовуваних на резервуємому ЦОДі.

Вимоги до приміщень і інженерних систем

Даний розділ розглядає вимоги до приміщень, у яких розташовується серверне й мережне встаткування, а також до інженерних систем, які підтримують дані приміщення.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Загальні вимоги до приміщень

Серверні приміщення всіх гіперЦОД повинні задовольняти наступним загальним вимогам:

– Забороняється розміщати серверні приміщення в приміщеннях, оснащених більшою кількістю інженерних споруджень, які становлять потенційну небезпеку для встаткування.

– Забороняється розміщати серверні приміщення під приміщеннями їдальні, туалетів і інших приміщень, пов'язаних зі споживанням води.

– Щоб уникнути протічок води з даху забороняється розміщати серверні приміщення на останньому поверсі будинку.

– Серверна кімната повинна являти собою приміщення з обмеженим доступом, призначене для розміщення серверного встаткування.

– Конструкція серверної кімнати повинна відповідати наступним вимогам:

– Підтримувати необхідну безперервність робочих процесів.

– Підтримувати необхідна вага встаткування серверної кімнати.

– Захищати коштовне встаткування й дані.

– Фізичний доступ до серверної кімнати повинні мати тільки вповноважені співробітники ІТ-підрозділів і обслуговуючих організацій.

– Для обмеження фізичного доступу до серверної кімнати повинні використовуватися автоматизовані системи контролю доступу.

Залежно від рівня гіперЦОД, серверна кімната повинна бути оснащена:

– джерелом безперебійного живлення;

– системою кондиціонування;

– дизель – генератором;

– системою регулювання чистоти й вологості повітря;

– серверними й телекомунікаційними шафами, стійками. Висота шафи, що рекомендується, 42U;

– системами контролю стану внутрішнього середовища:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

- системою раннього димовизначення;
- датчиками доступу;
- датчиками фізичного стану встаткування; Дозволяється використовувати убудовані в устаткування датчики фізичного стану;
- датчиками температури/вологості;
- системою відеоспостереження.

У серверній кімнаті рекомендується мати фальшпол. Фальшпол є необхідним компонентом, тому що під нього нагнітається охолоджене повітря, під ним розташовуються кабелі електропостачання й слабкострумова інфраструктура. Рекомендується фальшпол із МДФ-плиток на металевій основі з ламинованим покриттям або знімний фальшпол з покриттям «керамогранит» розміром 600 x 600 мм. Висота над рівнем підлоги – від 100 до 800 мм, для серверних приміщеннях найбільше оптимально 350 – 500 мм. Для розподілу потоків холодного повітря від системи кондиціонування рекомендується використовувати перфоровані панелі.

Рекомендується при розрахунку площі приміщень виходити з розрахунку 2 кв. м на один 19-дюймова шафа, якщо іншого не передбачено технічним проектом або робочою документацією.

Структуровані кабельні системи

Структурована кабельна система (СКС) – фізична основа інформаційної інфраструктури підприємства, що дозволяє звести в єдину систему безліч інформаційних сервісів різного призначення.

СКС являє собою ієрархічну кабельну систему будинку або групи будинків, розділену на структурні підсистеми. Вона складається з набору мідних і оптичних кабелів, крос-панелей, сполучних шнурів, кабельних рознімачів, модульних гнізд, інформаційних розеток і допоміжного устаткування. Всі перераховані елементи інтегруються в єдину систему й експлуатуються відповідно до певних правил.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Кабельна система – це система, елементами якої є кабелі й компоненти, які пов'язані з кабелем. До кабельних компонентів ставиться все пасивне комутаційне встаткування, що служить для з'єднання або фізичного закінчення (термінування) кабелю – телекомунікаційні розетки на робочих місцях, кроссові й комутаційні панелі в телекомунікаційних приміщеннях, муфти й сплайси.

Загальні вимоги до СКС наступні:

- СКС повинна бути спроектована з надмірністю по кількості підключень.
- Робоче місце повинне мати, як мінімум, одне рознімання для підключення до ЛОМ і одне рознімання для підключення до телефонної мережі.
- Максимальна відстань горизонтальної проводки не повинне перевищувати 90м.
- Кабельні компоненти повинні бути не менш категорії 5е для піключення АРМ, або встаткування на комутатори рівня доступу й не менш категорії 6 або оптичних підключень для підключення комутаторів рівня доступу до комутаторів рівня розподілу й далі.
- У всіх типах гіперЦОД також повинні використовуватися кабельні компоненти не менш категорії 6.
- Прокладку кабелів у коридорах повинна здійснюватися за фальшпотолком, якщо такий є, а при його відсутності – у спеціалізованих кабель-каналах (коробах) або в існуючим закладних. у робочих приміщеннях підвід кабелю до робочих місць виробляється в кабельканалах.
- СКС повинні бути документовані.
- На СКС повинна надаватися гарантія виробника на працездатність на строк не менш 25 років. Підрядник повинен бути сертифікований виробником і мати перевірене вимірювальне встаткування. При проведенні приймальних випробувань підрядник повинен надати протоколи тестування СКС на відповідність установленим нормам.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Електропостачання

Загальні вимоги до системи електропостачання всіх гіперЦОД наступні:

– У серверне приміщення електроживлення повинне подаватися від головного щита будинку, де б дане приміщення не перебувало. Також від головної шини заземлення будинку проводиться кабель заземлення до контуру заземлення серверного приміщення. Всі проведення повинні мати відповідний перетин відповідно до технічного проекту й колір, відповідно до нормативних документів.

– Живлення ПК, периферійних пристроїв, офісної техніки, серверів, систем зберігання й активного мережного встаткування повинне бути відділене від живлення промислових установок. Живлення повинне здійснюватися від окремих поетажних автоматів, а ті, у свою чергу, окремо повинні приєднуватися до головного щита будинку через окрему систему автоматів.

– При організації живлення ПК, периферійних пристроїв і офісної техніки рекомендується після автоматів у поетажних щитах установлювати пристрою захисного відключення (ПЗВ) відповідно до діючих норм.

– Систему електропостачання для гіперЦОД I рівня варто організувати від двох територіально рознесених трансформаторних підстанцій. Кабельні лінії повинні йти незалежними маршрутами. Потрібно використовувати автомати вибору резерву (АВР), що здійснюють вибір і перемикання між основними й резервними лініями.

– Для гіперЦОД I рівня необхідно, а для гіперЦОД II рівня рекомендується використовувати дизель-генераторні електростанції (ДГЕ). У схемі електропостачання вони повинні розташовуватися паралельно введенням кабелів електроживлення в будинок. Для правильної роботи ДГЕ і двох незалежних уведень повинен бути передбачений пристрій автоматичного включення резервного живлення. У випадку повного провалля електроживлення, або невідповідності його необхідним параметрам (напруга, частота, "чистота") повинен здійснюватися автоматичний запуск ДГЕ, і навантаження переводиться

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

на неї. ДГЕ повинна мати запас палива, розрахований не менш чим на 8 годин безперервної роботи й можливість поповнення паливом без зупинки генератора. ДГЕ повинні мати можливість безперервної роботи до 3 місяців за умови налагодженої поставки палива.

– Для гіперЦОД I і II рівнів після введення кабелів електроживлення в будинок або після ДГЕ, при її наявності, повинні бути встановлені централізовані джерела безперебійного живлення (ДБЖ) подвійного перетворення.

Вимоги до електропостачання шаф для гіперЦОД всіх рівнів:

– До кожної шафи повинне бути підведене живляча напруга $220 \pm 10\%$ У змінного струму від двох незалежних джерел через індивідуальні автоматичні вимикачі.

– Підключення встаткування, що має два блоки живлення, здійснювати до двох незалежних джерел. Підключення встаткування, що має один блок живлення, здійснювати до одного із джерел живлення, рівномірно розподіляючи навантаження відповідно до енергоспоживання, зазначеним у паспорті встаткування.

– Споживана потужність повинна бути розрахована в технічному проекті. Якщо встановлюються порожні шафи й на розміщення встаткування в них ще немає проекту, то рекомендується оцінювати енергоспоживання в середньому 4 кВт на шафу. Якщо в шафах передбачається встановлювати blade-сервери або інше встаткування, що має підвищене енергоспоживання, то споживана потужність повинна бути скоректована відповідно до документації виробника.

Кондиціонування й система холодопостачання

Вимоги до систем кондиціонування й холодопостачання для всіх гіперЦОД:

– Серверні приміщення повинні бути обладнані промисловою прецизійною системою кондиціонування й вентиляції (системи холодопостачання).

– У завдання системи холодопостачання повинне входити підтримка усередині приміщення робочої температури в межах від 19 до 24 °С и вологості від 40 до 80%.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

– Резервування системи холодопостачання гіперЦОД I рівня обов'язково, а для гіперЦОД II рівня рекомендується здійснювати за схемою з N+1 (з одним запасним кондиціонером). Всі кондиціонери повинні бути підключені до єдиної системи керування. Програмне забезпечення повинне дозволяти здійснювати ротацію запасного кондиціонера, що дозволяє більш ефективно витратити ресурс системи холодопостачання в цілому.

– Для гіперЦОД I рівня необхідно, а для гіперЦОД II рівня рекомендується організувати приплив свіжого повітря з вулиці. Приплив рекомендується здійснювати через спеціальну установку, що підготовляє вуличне повітря. Крім того, вона повинна створювати усередині приміщення додатковий тиск, що перешкоджає проникненню усередину пилу.

– Для зволоження повітря в гіперЦОД I і II рівнів рекомендується використовувати парогенератори. Сухе повітря малоефективне для охолодження системою хладопостачання в силу фізичних принципів кондиціонування. При зниженні вологості електростатичний потенціал збільшується, що може бути причиною виводу встаткування з ладу.

– Рекомендується вивід гарячого повітря із шаф у повітряний простір і його транспортування до кондиціонера, або розглянути можливість організації холодних і гарячих коридорів, або передбачити використання кондиціонерів, розташованих між стійок.

– При використанні системи кондиціонування з воздуховодами й забором гарячого повітря зверху шафи необхідна наявність системи примусової вентиляції у верхній частині шафи.

– При використанні системи кондиціонування без воздуховодів необхідно використовувати стійки з перфорованими передніми й задніми дверима для кращого охолодження від системи кондиціонування.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Системи раннього виявлення пожежі й пожежогасіння

Вимоги до системи раннього виявлення пожежі й газового пожежогасіння для всіх гіперЦОД:

– гіперЦОД повинні бути обладнані системою автоматичного пожежогасіння (ДСТУ 12.1.004-91.ССБТ). Система пожежогасіння не повинна завдавати шкоди встаткуванню.

– Система газового пожежогасіння повинна спрацювати в зародковій фазі розвитку пожежі, тобто коли відбувається тління елементів, що нагріваються, або початкове запалення, і за час менш однієї мінути згасити вогнища загоряння.

– Комплекс попередження про пожежу й пожежогасіння повинен повідомити про потенційну можливість загоряння набагато раніше, ніж прийде задіяти систему гасіння. Це повинне бути досягнуте установкою великої кількості високочутливих димових, оптичних, хімічних, спектральних і інших пожежних оповіщувачів, ув'язаних у єдину інтелектуальну систему оповіщення про пожежу й пожежогасіння, а також комплексом організаційних заходів. У нього повинен входити постійний візуальний огляд устаткування, дотримання пожежних норм і правил, а також правил експлуатації електроустановок.

– Рекомендується використовувати вогнетушачі суміші на основі хладонів або інертних газів, тому що вони завдають найменшої шкоди встаткуванню.

– Потрібно передбачити систему видалення газу із приміщення після спрацювання системи пожежогасіння.

– При спрацюванні системи газового пожежогасіння повинні відключатися всі системи нагнітаюче повітря в приміщення гіперЦОД.

Комплексні системи безпеки

Комплексні системи безпеки повинні складатися з:

- системи відеоспостереження;
- системи розмежування фізичного доступу.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

нею своїх завдань (системне й прикладне програмне забезпечення, бази даних і інші набори даних), а також архіви, журнали транзакцій, системні журнали й т.д.

– Резервному копіюванню підлягають всі настроювання активного мережного встаткування.

– Резервному копіюванню підлягає вся проектна документація (технічний проект, робоча документація, експлуатаційна документація).

– Всі програмні засоби, використовувані в системі повинні мати еталонні (дистрибутивні) копії. Їхнє місцезнаходження й відомості про особи, відповідальних за їхнє створення, зберігання й використання повинне бути зазначене явно у відповідних документах. Там же повинні бути зазначені переліки наборів даному, підлягаючому страховому копіюванню, періодичність копіювання, місце зберігання й відповідальні за створення, зберігання й використання страхових копій даних.

– Необхідні дії персоналу по створенню, зберіганню й використанню резервних копій програм і даних повинні бути відбиті у функціональних обов'язках відповідних категорій персоналу.

– Кожний носій, що містить резервну копію, повинен мати мітку, що містить дані про клас, цінність, призначення збереженої інформації, відповідальному за створення, зберігання й використання, дату останнього копіювання, місце зберігання та інше.

Вимоги до системи керування й моніторингу

Даний розділ розглядає вимоги до системи керування й моніторингу.

Загальні вимоги

Завдання системи керування й моніторингу:

- Підвищення ефективності використання ІТ-інфраструктури.
- Підтримка високого рівня обслуговування прикладних систем.
- Превентивне рішення потенційних проблем.
- Скорочення втрат через простої при відновленні даних.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

- Віддалений моніторинг об'єктів керування.
- Моніторинг контрольованих об'єктів за допомогою агентів.
- Вибір параметрів моніторингу й настроювання порогів спрацьовування агентів, для оцінки поточного стану систем.
- Централізована реєстрація подій, що відбуваються в контрольованих об'єктах СПД, операційних системах, СУБД, додатках, інформаційних сервісах.
- Розширення списку реєструємих подій і адаптація до використовуваних додатків і існуючих технологій.
- Централізована обробка всіх реєструємих подій.
- Оповіщення операторів системи про роботу інформаційних ресурсів за допомогою видачі інформаційного повідомлення на консоль оператора.
- Сповіщати операторів системи про виниклі проблеми за допомогою видачі звукового сигналу.
- Аналіз продуктивності роботи об'єктів керування.
- Автоматична обробка й графічне подання оперативної інформації зі стану інформаційних сервісів.
- Збір, зберігання й аналіз параметрів функціонування об'єктів керування.

Вимоги до функціональності СУМ гіперЦОД III рівня

СУМ гіперЦОД III рівня повинна забезпечувати виконання наступних функцій:

- моніторинг основних параметрів елементів ІТ інфраструктури.
- виявлення й локалізація відмов елементів ІТ інфраструктури.
- Система моніторингу повинна інтегруватися із системами моніторингу гіперЦОД I і II рівнів.

Вимоги до керування й моніторингу мультисервісної мережі

Моніторинг ІР-мереж, керування конфігураціями, збоями, продуктивністю, а також методи й засоби інвентаризації ІР-мереж повинні бути строго регламентовані й автоматизовані.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Відповідна база даних для гіперЦОД I і II рівнів повинна містити повну й достовірну інформацію про всі елементи мережі в ієрархічному виді з урахуванням географічної ієрархії з однієї сторони й функціональної ієрархії (додатка, IP-мережі, базові мережі) з іншої.

Функціонально база даних може бути представлена як дві взаємодіючі бази даних: інвентаризаційна й системи керування подіями в мережі.

Інвентаризаційна база даних повинна поєднувати інформацію від різних джерел і надавати її в зручній формі. Рекомендований набір інформації для включення в інвентаризаційну базу даних:

- Про топологію мережі й установлене встаткування в керованих мережах.

- Докладну інформацію про використовувані канали зв'язку (у випадку орендованих каналів – інформацію про організації, що надала канал, контактні дані фахівців і служби технічної підтримки, способах зв'язку).

- Повну специфікацію встаткування: поточну версію ПЗ, заводські й інвентаризаційні номери, файли конфігурації, версію ПЗ, контактну інформацію обслуговуючої організації, місце установки й відповідальні особи.

- Для мережного встаткування – докладний опис інтерфейсів у табличному виді із вказівкою IP-адрес, VLAN, підключених мереж або серверів додатків для LAN-інтерфейсів, або використовуваних каналів зв'язку (фізичних і віртуальних), протоколів маршрутизації й підключеного віддаленого встаткування для WAN-інтерфейсів.

Автоматизована система обробки подій для гіперЦОД I і II рівнів повинна забезпечувати:

- Автоматизований збір у режимі реального часу й зберігання інформації про збої, несправності, перевищення критичних порогів і т.п. активного встаткування й каналів зв'язку.

- Повідомлення обслуговуючого персоналу про виникаючі проблеми й передачу цієї інформації з ієрархічної структури (адміністративної, топологічної, системної) відповідно до встановлених адміністративних правил.
- Відображення історії обробки події (ким, коли, які дії були початі).
- Збереження історії подій по кожному об'єкті керування.
- Прив'язку події до об'єкта, тобто в інвентаризаційній складовій повинна бути посилання на історію подій об'єкта й навпаки.

Крім цього, система обробки подій повинна надавати аналітичну інформацію відповідно до заданих адміністративних вимог (наприклад, вибірку по об'єктах, на яких не були вчасно проведені профілактичні роботи, вибірку по подіях, які не були закриті протягом місяця й т.п.)

Функціональна схема розробленої системи зображена на рисунку 3.2. Існує не занадто багато способів управління навантаженням трафіку гіперЦОД. Найпростіший з них – збільшення смуги пропускання мережі гіперЦОД за рахунок нарощування апаратних можливостей устаткування. Можна використовувати й такі прийоми, як завдання пріоритетів даних, організація черг, запобігання перевантажень і формування трафіку гіперЦОД. Керування мережею за заданими правилами в перспективі повинне об'єднати всі ці способи в єдину автоматизовану систему, що буде гарантувати якість послуг абсолютно на всіх ділянках мережі гіперЦОД.

Збільшення апаратної потужності, безсумнівно, є найбільш ефективним засобом управління навантаженням трафіку гіперЦОД у корпоративній мережі гіперЦОД. Тиск із боку конкурентів, необхідність підвищення ефективності виробництва, поява нових технологій, що дозволяють оснащувати спеціалізовані мікросхеми (ASIC) найрізноманітнішими функціями, – все це змушує постачальників комутаційного встаткування для корпоративних мереж викидати на ринок усе більше швидкодіючі пристрої за цінами, порівняним з вартістю моделей колишнього покоління.

Малоймовірно, що в доступному для огляду майбутньому даний підхід до управління навантаженням трафіку гіперЦОД у корпоративних мережах

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

перестане бути пріоритетним. Оскільки в корпоративних мережах вдається забезпечити гарантовану якість послуг, не прибігаючи до дорогої модернізації усього встаткування й серйозних змін у системі керування мережею, мережні адміністратори будуть звертати увагу й на програмні засоби, що дозволяють реалізувати управління навантаженням трафіку гіперЦОД.

Отже, найбільше поширення, швидше за все, одержить комбінований підхід. Деякі виробники висловлюються на його користь, затверджуючи, що найкраще збільшувати пропускну здатність мережі гіперЦОД не прямо, а за рахунок інтелектуальних можливостей устаткування, що має засоби забезпечення управління навантаженням трафіку гіперЦОД. Правда, виробники мережних пристроїв навряд чи можуть бути об'єктивними в цьому питанні, тому що вони зацікавлені в збуті тих самих продуктів, які підтримують гарантовану якість послуг.

У глобальних мережах нарощування апаратних потужностей використовується рідше. Звичайно, зниження вартості смуги пропускання зробило б передачу даних по глобальних мережах доступною для більше широкого кола користувачів (і навіть трохи знизило б актуальність впровадження гарантованої якості послуг). Але в найближчому майбутньому вартість смуги пропускання в глобальних мережах буде залишатися досить високою, тому й нарощування апаратної потужності не стане настільки популярним, як у корпоративних мережах.

З рисунку видно, що розроблена система складається з наступних функціональних частин:

- Блок інтерфейсу користувача.
- Блок моніторингу мережі гіперЦОД.
- Блок дослідження можливостей механізмів WRED.
- Блок дослідження можливостей механізмів WFQ.
- Блок призначення пріоритетів.
- Блок організації та обслуговування черг.
- Блок управління навантаженням.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

- Блок формування трафіку гіперЦОД.
- Блок визначення параметрів управління навантаженням трафіку гіперЦОД.
- Блок примусового завдання параметрів управління навантаженням трафіку гіперЦОД.
- Блок моделювання завантаженості мережі гіперЦОД.

Розглянемо ці блоки більш детально.

Блок інтерфейсу користувача

Призначений для реалізації взаємодії користувача, або дослідника з системою.

Блок моніторингу мережі гіперЦОД

Призначений для аналізу поточного стану мережі гіперЦОД.

Блок дослідження можливостей механізмів WRED

Одним з методів управління навантаженням трафіку гіперЦОД, призначених для забезпечення необхідних вимог до різних потоків даних – запобігання перевантажень (congestion avoidance). Він заснований на обмеженні розмір черги, сигналізуючи джерелам даних про необхідність зменшити швидкість передачі інформації (WRED – Weighted ranDoSm early detection).

Блок дослідження можливостей механізмів WFQ

Другим з методів управління навантаженням трафіку гіперЦОД, призначених для забезпечення необхідних вимог до різних потоків даних – керування перевантаженням (congestion management). Він заснований на присвоєнні квот і пріоритетів потокам, і у випадку перевантаження, потоки одержують якість, обмежену їхньою квотою й пріоритетом (WFQ – Weighted Fair Queuing).

Блок призначення пріоритетів

Нарівні з нарощуванням апаратного забезпечення мережі гіперЦОД для реалізації управління навантаженням трафіку гіперЦОД застосовуються й засоби типу завдання пріоритетів даних і організації черг. Маршрутизатори підтримують ці механізми протягом багатьох років, як і деякі з нових комутаторів для каналів

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Gigabit Ethernet. Однак ПЗ для керування мережею за заданими правилами, яких необхідно для практичного втілення цієї технології, поки не розроблено. Серед нових комутаторів такого класу можна назвати CoreBuilder 3500, CoreBuilder 9000 і SuperStack II компанії 3Com, пристрою серії Accelar фірми Bay Networks, SmartSwitch Router компанії Cabletron Systems, а також Catalyst 5000 і Catalyst 8000 виробництва Cisco.

Способи пріоритезації даних можна умовно підрозділити на явні й неявні.

При неявному призначенні пріоритетів маршрутизатор або комутатор автоматично привласнює послугам відповідні рівні, виходячи із заданих адміністратором мережі гіперЦОД критеріїв (наприклад, типу додатка для застосовуваного протоколу передачі або адреси джерела). Кожний вхідний пакет аналізується (фільтрується) на відповідність цим критеріям. Механізм неявної пріоритезації підтримують практично всі маршрутизатори.

Деякі комутатори теж здатні задавати пріоритети, але мають обмежений набір функцій. Так, комутатори можуть забезпечувати пріоритезацію даних по типу віртуальної корпоративної мережі гіперЦОД, адресі джерела або адресата, але не використовують інформацію більш високого рівня (протокол передачі або тип додатка). Розроблювальні в цей час системи керування мережею за заданими правилами дозволяють реалізувати більше зроблені схеми пріоритезації даних при роботі з такими комутаторами.

При явній пріоритезації даних користувач або додаток запитує певний рівень служби, а комутатор або маршрутизатор намагається задовольнити запит. Імовірно, самим популярним механізмом явної пріоритезації стане протокол IP Precedence (протокол старшинства), що одержав другу назву IP TOS (IP Type Of Service), – один з розділів четвертої версії протоколу IP.

IP TOS резервує спеціальне поле в заголовку пакета, де можуть бути зазначені ознаки управління навантаженням трафіку гіперЦОД, що визначають час затримки, швидкість пропущення й рівень надійності передачі пакета. Однак знайдеться небагато популярних додатків – за винятком мультимедійного ПЗ, – у які реалізована підтримка протоколу IP TOS.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Зараз розробляється протокол резервування ресурсів RSVP, що передбачає більше складний, чим в IP TOS, механізм передачі від додатка до маршрутизатору запиту на гарантовану якість послуг. Як і IP TOS, протокол RSVP поки не одержав широкої підтримки розроблювачів – він реалізований лише в окремих типах маршрутизаторів. Поширення RSVP стримується через те, що не вирішені деякі питання, пов'язані із сумісністю різних мереж. До того ж застосування RSVP значно збільшує навантаження на маршрутизатори й може привести до зниження швидкодії цих пристроїв.

Видимо, у доступному для огляду майбутньому неявна пріоритезація, не потребує серйозних обчислювальних потужностей маршрутизатора, залишиться більше популярною, чим явна. Крім того, при явному завданні пріоритетів значно ускладнюється керування мережею. Кінцеві користувачі, швидше за все, будуть набувати своє програмне забезпечення на запит найвищого з можливих рівнів послуг. Відповідно, адміністраторові мережі гіперЦОД прийдеться розробляти правила керування користувачами й, можливо, навіть побудувати служби з гарантованою якістю для кожного користувача окремо.

Блок організації та обслуговування черг

Після того як переданим по мережі гіперЦОД даним призначені відповідні пріоритети (за допомогою явних або неявних методів), потрібно визначити порядок передачі цих даних, задавши алгоритм обслуговування черг із необхідною якістю (рівнем управління навантаженням трафіку гіперЦОД). По суті, черги являють собою області пам'яті комутатора або маршрутизатора, у яких групуються пакети з однаковими пріоритетами передачі. Алгоритм обслуговування черги визначає порядок, у якому відбувається передача пакетів, що зберігаються в ній. Зміст застосування всіх алгоритмів зводиться до того, щоб забезпечити найкраще обслуговування трафіку гіперЦОД з більш високим пріоритетом за умови, що й пакету з низьким пріоритетом гарантується відповідна увага.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

При використанні способів завдання явних і неявних пріоритетів алгоритм обробки черг визначає порядок їхнього обслуговування. Відповідно до цього алгоритму на кожні два пакети, переданих у мережу із черги 1 (з високим пріоритетом) доводиться по одному пакету із черг 2 і 3. Пакети з однаковими пріоритетами передаються за принципом FIFO (“першим прийшов – першим вийшов”).

Якщо в мережі гіперЦОД виникає перевантаження, служба черг не гарантує своєчасного досягнення пункту призначення найбільш важливими даними. Гарантується лише те, що ці пакети будуть передані раніше, ніж ті, що мають більш низький пріоритет.

Сучасні служби управління навантаженням трафіку гіперЦОД вирішують таке завдання за рахунок резервування смуги пропускання. Кожній із черг (або їхніх груп) виділяється заздалегідь задана величина смуги пропускання, що гарантує певну смугу пропускання для черги з більш високим пріоритетом. Для критичних ситуацій, коли обсяг даних у черзі перевищує розміри смуги пропускання, в алгоритмах обслуговування звичайно передбачається передача трафіку гіперЦОД з високим пріоритетом на смугу пропускання, “яка належить” чергам з низьким пріоритетом, і навпаки. Найпростіші алгоритми обслуговують кожен чергу за принципом FIFO. При цьому передача кадрів великого розміру, що мають високий пріоритет, може приводити до затримок трафіку гіперЦОД іншого додатка з настільки ж високим пріоритетом, але меншим обсягом.

У більш складних алгоритмах уживає спроба “справедливої” обробки черг. Наприклад, алгоритм рівномірного пропорційного (або зваженого) обслуговування (WFQ – Weighted Fair Queuing), розроблений компанією Cisco, підрозділяє додатки на потребуючі великої й малої ширини смуги пропускання, а сама смуга пропускання розподіляється між всіма додатками нарівно. Слід зазначити, що основні виробники маршрутизаторів самі розробляють алгоритми обслуговування черг і використовують для їхнього опису власну термінологію.

Істотним недоліком сучасних маршрутизаторів і комутаторів є те, що вони підтримують мале число черг. Найчастіше виробники організують служби управління навантаженням трафіку гіперЦОД, що використовують чотири черги, хоча чим більше черг, тим більше різних пріоритетів можна привласнити переданим пакетам і тим “більш справедливо” розподілити смугу пропускання між додатками. Наприклад, адміністратор у стані задати пріоритети таким чином, щоб перевага при передачі віддавалося пакетам, адресованим на більше віддалені вузли.

Блок управління навантаженням

Служба управління навантаженням трафіку гіперЦОД дає можливість використовувати для керування мережею два важливих механізми – керування в умовах перевантаження й запобігання перевантажень. Перший з них дозволяє кінцевій станції відразу знижувати швидкість передачі даних, коли в мережі гіперЦОД починається втрата пакетів. У протоколах TCP/IP і SNA цей механізм підтримується вже протягом декількох років. І хоча сам по собі він не гарантує якості передачі, при його використанні разом з механізмом запобігання перевантажень результати виявляються набагато кращими. У мережах TCP/IP механізм запобігання перевантажень застосовується досить давно, але лише в останні роки він стає стандартом “де-факто” для маршрутизаторів телекомунікаційних мереж і Internet.

Стандартним способом запобігання перевантажень у мережі гіперЦОД стало застосування механізму випадкового виділення пакетів (RanDoSm Early Detection, RED). При заповненні черг вище певної критичної оцінки цей механізм змушує маршрутизатор вибирати із черги за випадковим законом деякі пакети й “втрачати” їх. Швидкість передачі даних станціями-відправниками знижується, що й дозволяє уникнути переповнення черги.

Механізм пропорційного випадкового виділення пакетів – WRED (Weighted RED) – можна вважати наступною, більше зробленою “версією” RED. Він передбачає, що вибір пакетів, які повинні “втратитися”, буде відбуватися з обліком їх пріоритезації згідно IP TOS.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

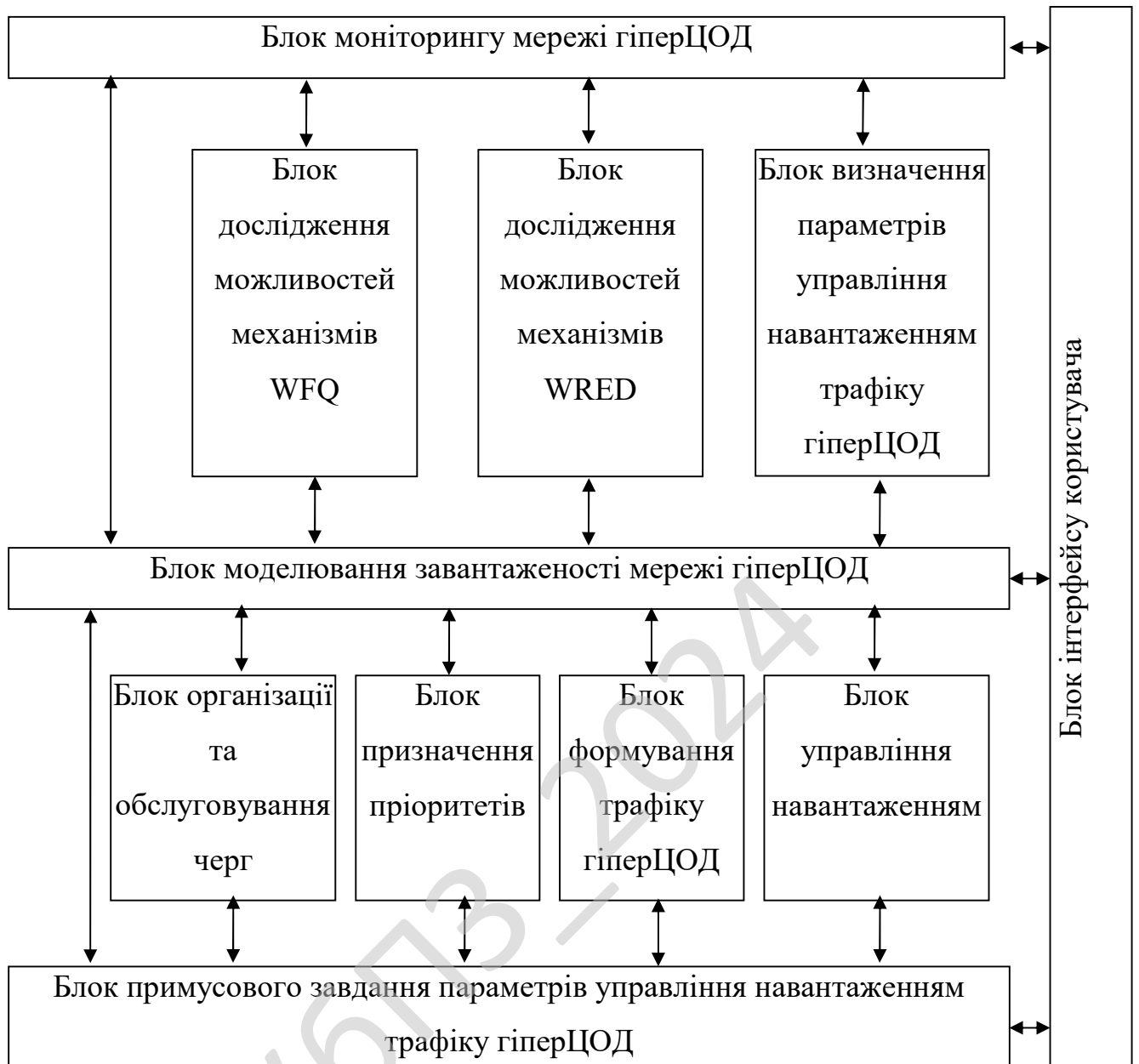


Рисунок 3.2 – Функціональна схема системи

Блок формування трафіку гіперЦОД

Формування трафіку гіперЦОД – це загальний термін, яким прийнято позначати різні способи маніпулювання даними для підвищення якості їхньої передачі. Один їхній таких способів – сегментація пакетів. У мережах АТМ гарантовано високий рівень управління навантаженням трафіку гіперЦОД досягається в тому числі й за рахунок малого розміру переданих пакетів

(осередків – у термінології АТМ). Максимальний час затримки при передачі будь-якого пакета мережі гіперЦОД АТМ – це час передачі одного осередку.

Запозичаючи корисні механізми технології АТМ, виробники маршрутизаторів і комутаторів починають забезпечувати у своїх продуктах можливість сегментації пакетів. Деякі пристрої, призначені для мереж frame relay, сегментують пакети, передані по каналах глобальних мереж, щоб гарантувати конкретний час передачі й мінімізувати затримки.

Ще один спосіб формування трафіку гіперЦОД – його “вирівнювання”. Для таких протоколів, як наприклад, AppleTalk, характерна нерівномірна передача пакетів, що часом приводить до появи в мережі гіперЦОД послідовностей або ланцюжків пакетів, а отже – до її перевантаження. Процедура вирівнювання трафіку гіперЦОД дозволяє розчленувати ланцюжки шляхом розміщення пакетів у буфері перед їхньою передачею в мережу. Для забезпечення більше рівномірної передачі даних можна також вирівнювати трафік кінцевих вузлів мережі гіперЦОД.

Блок визначення параметрів управління навантаженням трафіку гіперЦОД

Призначений для визначення існуючих параметрів якості обслуговування (управління навантаженням трафіку гіперЦОД). До них відносяться:

– Bandwidth (BW) – смуга пропускання, описує номінальну пропускну здатність середовища передачі інформації, визначає ширину каналу. Вимірюється в bit/s (bps), kbit/s (kbps), mbit/s (mbps).

– Delay – затримка при передачі пакета.

– Jitter – коливання (варіація) затримки при передачі пакетів.

– Packet Loss – втрати пакетів. Визначає кількість пакетів, що відкидаються мережею під час передачі.

Блок примусового завдання параметрів управління навантаженням трафіку гіперЦОД

Призначений для примусового завдання одного, або декількох параметрів якості обслуговування (управління навантаженням трафіку гіперЦОД). До них відносяться:

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

– Bandwidth (BW) – смуга пропусення, описує номінальну пропускнуну здатність середовища передачі інформації, визначає ширину каналу. Вимірюється в bit/s (bps), kbit/s (kbps), mbit/s (mbps).

– Delay – затримка при передачі пакета.

– Jitter – коливання (варіація) затримки при передачі пакетів.

– Packet Loss – втрати пакетів. Визначає кількість пакетів, що відкидаються мережею під час передачі.

Блок моделювання завантаженості мережі гіперЦОД

Призначений для моделювання завантаженості мережі гіперЦОД з визначеним трафіком та заданими параметрами якості обслуговування (управління навантаженням трафіку гіперЦОД).

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

3.4 Розробка діаграми процесів

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

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

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

Діаграма процесів розробленої системи зображена на рисунку 3.3.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю система управління гіперЦОД.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

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

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

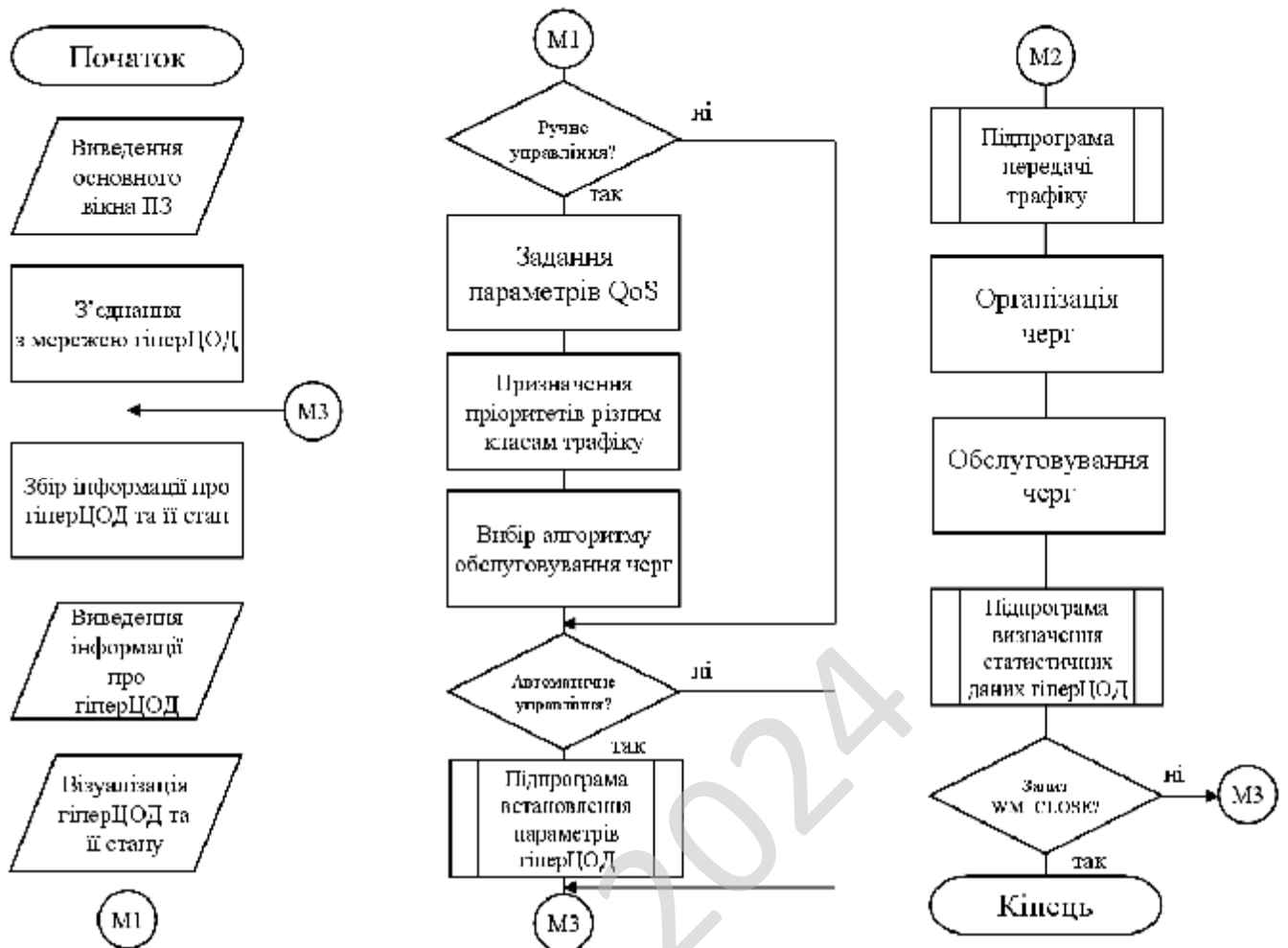


Рисунок 4.1 – Блок-схема основної програми

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

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

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

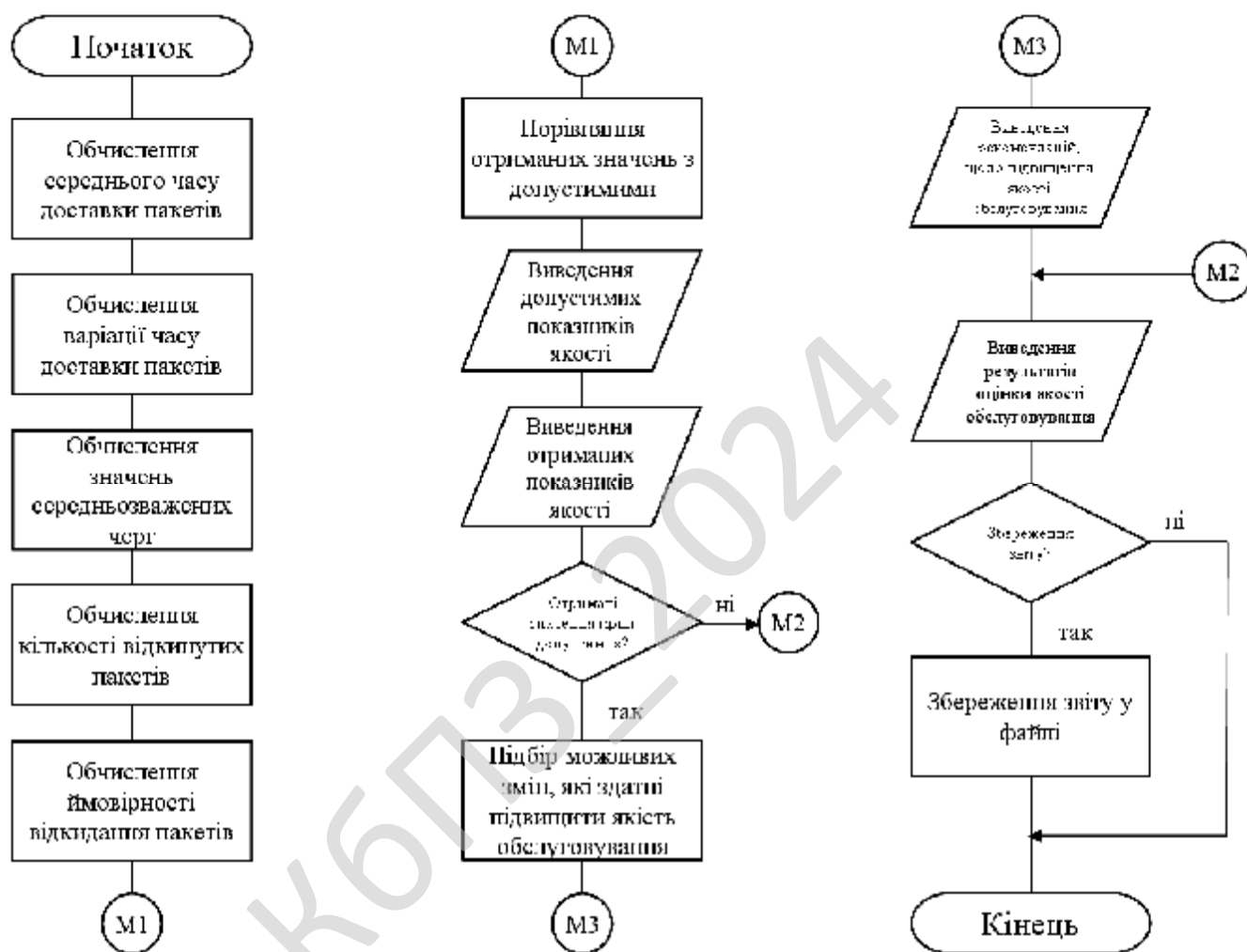


Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

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

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

Функціональні можливості:

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

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

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

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

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

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

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

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

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

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

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

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

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 браузерями.

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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

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

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

Хоча я реалізовував програму сам, було використано підходи Scrum для саморозвитку та пришвидшенню розробки, розглянемо цей метод. Scrum – підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки.

Підхід вперше описали Гіротака Такеучі та Ікуджіро Нонака в статті The New New Product Development Game (Гарвардський Діловий Огляд, січ–лют 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос-функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році ДеГрейс та Шталь у книжці Злі проблеми, справедливі рішення послалися на цей підхід, як на Scrum (штовханина; сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Такеучі і Нонака. Кен Швабер на початку 1990–х використовував підхід який привів Scrum в його компанію.

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

років працювали разом щоб обробити та описати весь їхній досвід та найкращі практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці Agile Software Development with SCRUM. Не зважаючи на те, що для Scrum нарікли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams), або як підхід управління розробкою і супроводом програм: Scrum of Scrums.

Scrum – це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи – ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін, та Команду, яка включає розробників.

Протягом кожного спринту, 15–30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, що повинна бути виконана.

Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту. Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

Product backlog – це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називається «історіями» (user story) або елементами backlog–у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum–процесу.

Обов'язкові поля:

1. ID – унікальний ідентифікатор, порядковий номер, який використовується для ідентифікації історій у разі їх перейменування.

2. Назва (Name) – стислий опис історії. Він повинен бути однозначним, щоб і розробники і product owner могли зрозуміти, про що йдеться і відрізнити одну історію від іншої.

3. Важливість (Importance) – ступінь важливості даної історії на погляд product owner 'а. Зазвичай являє собою натуральне число, іноді для цієї цілі використовуються числа Фібоначчі. Чим більше значення, тим більше пріоритет.

4. Попередня оцінка (initial estimate) – початкова оцінка об'єму робіт, необхідного для реалізації історії порівняно з іншими історіями. Вимірюється у story point'ах. Приблизно відповідає числу «ідеальних людино–днів».

5. Як продемонструвати (how to demo) – стисле пояснення того, як завершена задача буде продемонстрована у кінці спринта. Дане поле може являти собою код автоматизованого приймального тесту.

Додаткові поля. Іноді, також, використовуються додаткові поля у product backlog, в основному для того, щоб допомогти product owner'у визначитися з його пріоритетами.

Категорія (track). Наприклад, «панель управління» чи «оптимізація». За допомогою цього поля product owner може легко вибрати усі пункти категорії «оптимізація» і задати їм низький пріоритет.

Компоненти (components) – указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з

групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін.

Ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт.

ID у системі обліку помилок (bug tracking ID) – якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

Sprint backlog – містить функціональність, обрану Product Owner із Product Backlog. Всі функції розбиті по задачах, кожна з яких оцінюється командою. Кожен день команда оцінює об'єм роботи, який необхідно провести для завершення задачі.

Burndown chart – показує, скільки вже виконано і скільки ще залишається зробити.

Планування спринта (Sprint Planning Meeting)

Проходить на початку нової ітерації Спринта:

– Із Product Backlog обираються задачі, зобов'язання по виконанню яких за спринт приймає на себе команда.

– На основі обраних задач створюється Sprint Backlog. Кожна задача оцінюється у ідеальних людино-годинах.

– Рішення задачі не повинно займати більше 12 годин або одного дня. При необхідності задача розбивається на підзадачі.

– Обговорюється та визначається, яким чином буде реалізовано цей об'єм робіт.

– Тривалість наради обмежена зверху 4–8 годинами в залежності від тривалості ітерації, досвіду команди тощо.

– (перша частина наради) Беруть участь Product Owner + Команда: обирають задачі із Product Backlog.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

- Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення).
- Обмежена 1-3-ма годинами.

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою алгоритму хмарних технологій RSA. Спочатку необхідно обчислити пару ключів (секретний ключ і відкритий ключ). Для цього відправник електронних документів обчислює два більших простих числа P і Q , потім знаходить їхній добуток $N = P * Q$ і значення функції $\varphi(N) = (P-1)(Q-1)$. Далі відправник обчислює число E з умов $E < \varphi(N)$, НЗД($E, \varphi(N)$) = 1 і число D з умов $D < N$, $E * D \equiv 1 \pmod{\varphi(N)}$.

Пари чисел (E, N) є відкритим ключем. Цю пару чисел автор передає партнерам по переписці для перевірки його цифрових підписів. Число D зберігається автором як секретний ключ для підписування.

Допустимо, що відправник хоче підписати повідомлення M перед його відправленням. Спочатку повідомлення M (блок інформації, файл, таблиця) стискають за допомогою геш-функції $h(-)$ у ціле число m : $m = h(M)$.

Потім обчислюють цифровий підпис S під електронним документом M , використовуючи геш-значення m і секретний ключ D : $S = m \pmod{N}$.

Пари (M, S) передається партнерові-одержувачеві як електронний документ M , підписаний цифровим підписом S , причому підпис S сформований власником секретного ключа D .

Після прийому пари (M, S) одержувач обчислює геш-значення повідомлення M двома різними способами. Насамперед, він відновлює геш-значення m' , застосовуючи криптографічне перетворення підпису S з використанням відкритого ключа E : $m' = S^E \pmod{N}$.

Крім того, він знаходить результат гешування прийнятого повідомлення M з допомогою такої ж геш-функції $h(-)$: $m = h(M)$.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

Якщо дотримується рівність обчислених значень, тобто $S^E(\text{mod}) = h(M)$, то одержувач визнає пару (M, S) справжньою. Доведено, що тільки власник секретного ключа D може сформувати цифровий підпис S по документі M , а визначити секретне число D по відкритому числу E не легше, ніж розкласти модуль N на множники. Крім того, можна строго математично довести, що результат перевірки цифрового підпису S буде позитивним тільки в тому випадку, якщо при обчисленні S був використаний секретний ключ D , що відповідає відкритому ключу E . Тому відкритий ключ E іноді називають "ідентифікатором" того, хто підписав.

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ системи управління гіперЦОД яке зображено на рисунку 5.1. Розрахунки виконуються через командний рядок з наступною передачею результатів у графічний інтерфейс. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; Управління навантаженням; Моніторинг; Статистичні дані; Параметри; Довідка.
- Розділу обрання групи.
- Розділу виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

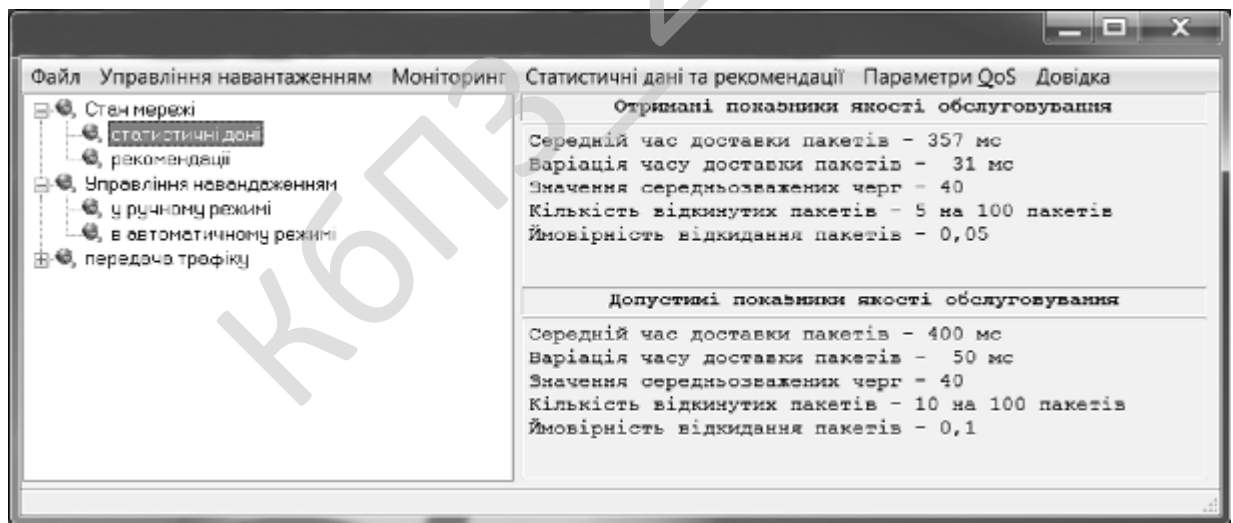


Рисунок 5.1 – Головне вікно ПЗ

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

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

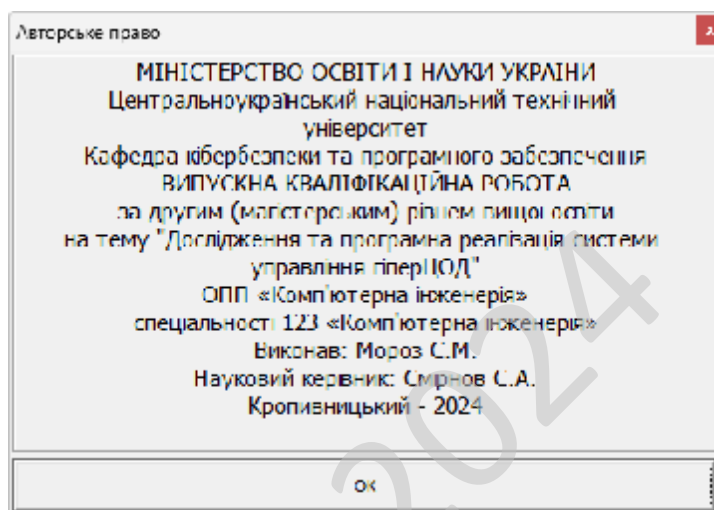


Рисунок 5.2 – Авторське право

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

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

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 в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

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

Ці тести демонструють:

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

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

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

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

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

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ).

Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєструватися), заплативши авторові певну суму.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

6 НАУКОВА НОВИЗНА

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

Метою розробки є дослідження та програмна реалізація системи управління гіперЦОД.

Об'єктом дослідження є процес управління гіперЦОД.

Предметом дослідження є методи управління гіперЦОД.

Методи дослідження базуються на методах хмарних технологій, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод управління гіперЦОД.
- Розроблено вітчизняний продукт управління гіперЦОД, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		120

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи управління гіперЦОД можуть бути цікаві різним групам зацікавлених осіб. Основні групи перераховано на рисунку 7.1.

ІТ-компанії
Клієнти, які використовують гіперконвергентні рішення
ІТ-менеджери та архітектори
Фахівці з кібербезпеки
Дослідники
Університети та коледжі
Інвестори в технологічні стартапи
Консультанти з оптимізації ІТ-інфраструктур
Органи, що регулюють ІТ-сектор
Користувачі, які безпосередньо користуються послугами
Журналісти та аналітики

Рисунок 7.1 – Цільова аудиторія

Таким чином, результати дослідження та програмної реалізації системи управління гіперЦОД можуть зацікавити широкий спектр учасників ринку, що може сприяти розвитку технологій і впровадженню нових рішень в бізнесі.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи управління гіперЦОД може бути здійснена за допомогою методів експертних оцінок.

Спочатку слід визначити критерії, які будуть використані для оцінки привабливості проекту. Це можуть бути критерії окреслені на рисунку 7.2.

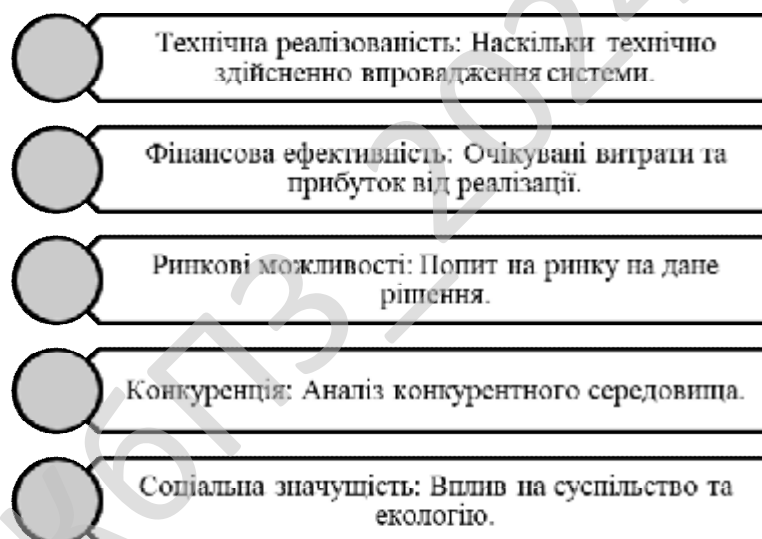


Рисунок 7.2 – Критерії експертної оцінки

Сформуємо групу експертів, які мають досвід у відповідних галузях (ІТ, бізнес, фінанси, маркетинг). Це : ІТ-архітектори, бізнес-аналітики, спеціалісти з продажу, фінансові аналітики. Експертам пропонується оцінити кожен критерій за шкалою, наприклад, від 1 до 5, де 1 – низька привабливість, а 5 – висока. Після цього результати підраховуються і виводиться середній по критерію: технічна реалізованість: 4, фінансова ефективність: 3, ринкові можливості: 5, конкуренція:

2, соціальна значущість: 4. Призначаються ваги кожному критерію в залежності від його важливості (наприклад, в сумі 1): технічна реалізованість: 0.25, фінансова ефективність: 0.30, ринкові можливості: 0.25, конкуренція: 0.10, соціальна значущість: 0.10. Обчислюється загальна оцінка привабливості проекту, використовуючи формулу: $\text{Загальна оцінка} = (\text{Оцінка1} \times \text{Вага1}) + (\text{Оцінка2} \times \text{Вага2}) + (\text{Оцінка3} \times \text{Вага3}) + (\text{Оцінка4} \times \text{Вага4}) + (\text{Оцінка5} \times \text{Вага5})$

$\text{Загальна оцінка} = (4 \times 0.25) + (3 \times 0.30) + (5 \times 0.25) + (2 \times 0.10) + (4 \times 0.10) = 1.00 + 0.90 + 1.25 + 0.20 + 0.40 = 3.75$ Загальна оцінка привабливості проекту становить 3.75 з максимально можливої оцінки 5. Це свідчить про те, що проект має високу привабливість для впровадження. Застосування експертних оцінок дозволяє об'єктивно оцінити привабливість програмної реалізації системи управління гіперЦОД, сприяючи прийняттю обґрунтованих рішень щодо інвестицій та подальшої реалізації проекту.

7.3 Вибір методу оцінки вартості ПЗ

Для програмної реалізації системи управління гіперЦОД найкраще використовувати метод оцінки на основі вартості життєвого циклу в поєднанні з методом аналогій. Це дозволить отримати більш точну оцінку вартості проекту з урахуванням всіх етапів його реалізації та експлуатації, а також зрозуміти, які витрати можуть виникнути в процесі його впровадження.

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

Приклад економічної ефективності від впровадження системи управління гіперЦОД може включати аналіз витрат та вигод, що дозволяє оцінити, наскільки вигідно інвестувати в дану технологію. На рисунку 7.4. зробимо розгорнутий перелік ключових економічних вигод, які можуть бути отримані в результаті впровадження системи управління гіперЦОД.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		123

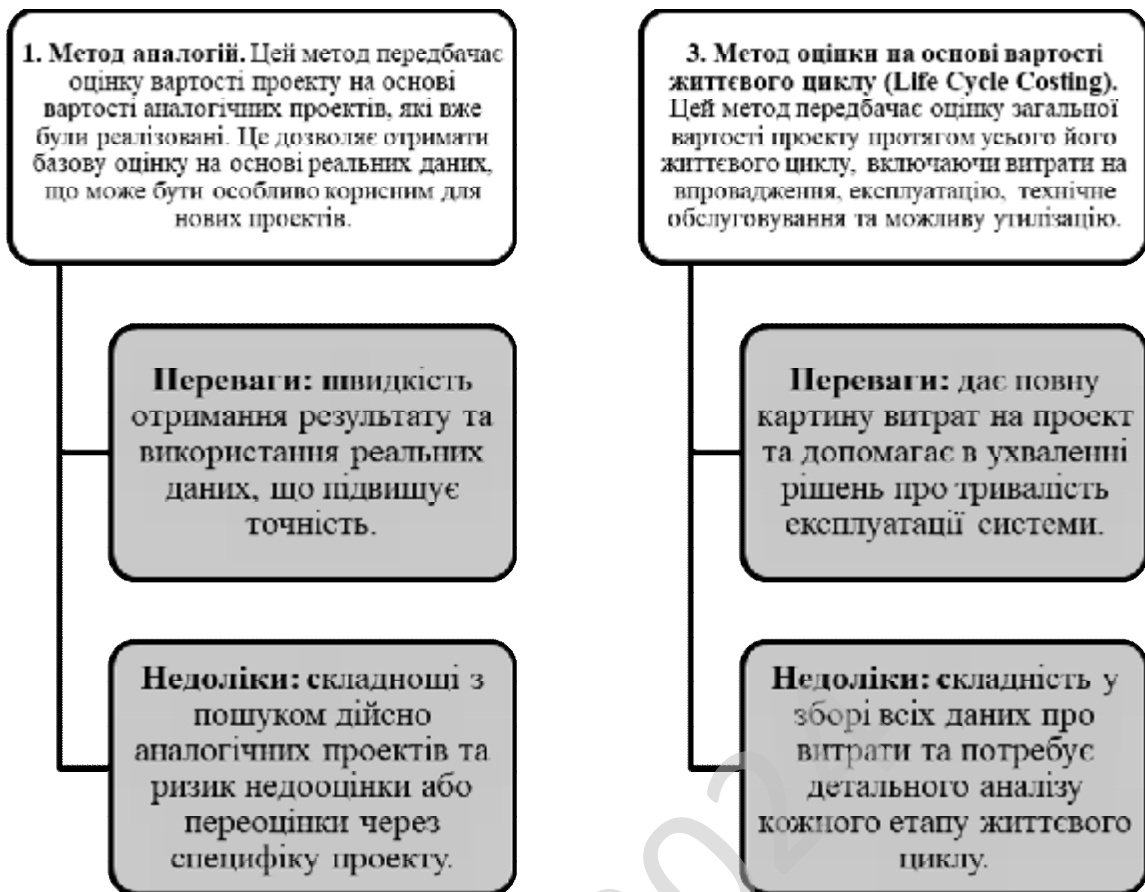


Рисунок 7.3 – Методи оцінки вартості



Рисунок 7.4 – Економічна ефективність від реалізації проекту для клієнта

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи управління гіперЦОД (гіперконвергентного дата-центра) мають включати пункти запропоновані на рисунку 7.6.

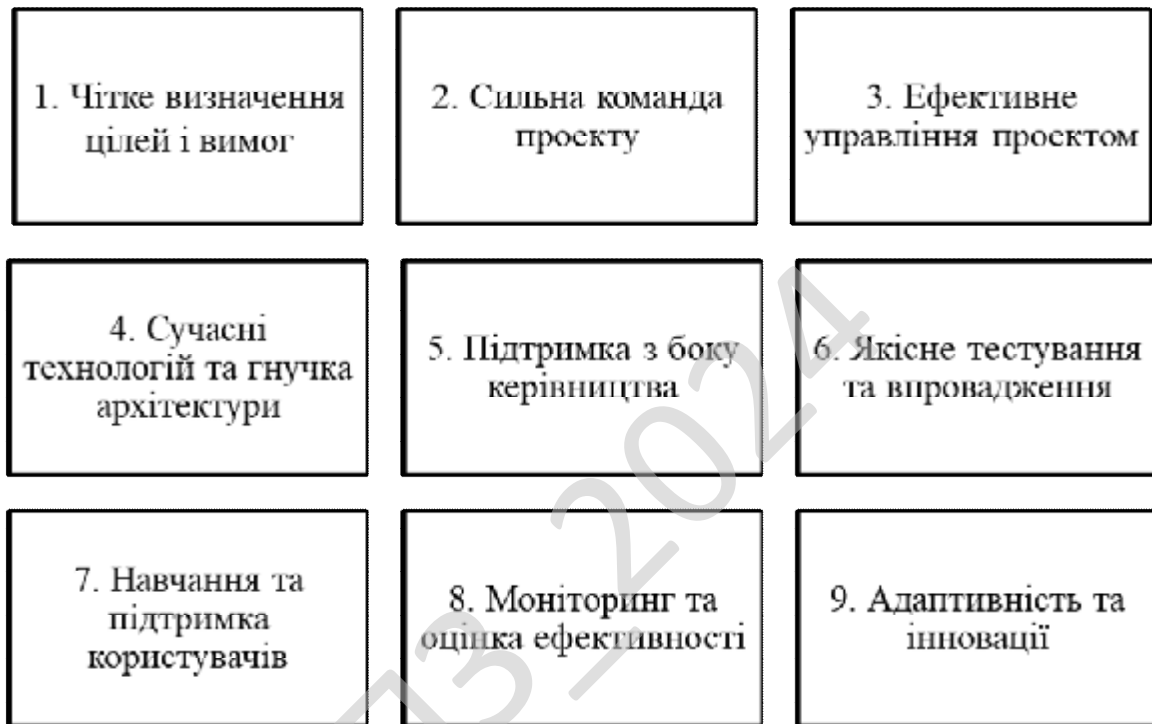


Рисунок 7.6 – Ключові фактори успіху проєкту

Успіх проєкту програмної реалізації системи управління гіперЦОД залежить від інтеграції всіх цих факторів. Чітке планування, ефективна команда, технологічні інновації та підтримка з боку керівництва – все це є критично важливим для досягнення позитивного результату.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Електронно-обчислювальна машина (ЕОМ) відіграє важливу роль у житті сучасної людини. Кожного дня мільйони людей використовують ЕОМ для пошуку необхідної інформації, спілкуванні у соціальних мережах, перегляду новин, роботи тощо. Багато людей користуються ЕОМ у професійних цілях, оскільки завдяки ЕОМ з'явилося багато нових професій.

Аналізуючи умови працівників ІТ-сфери, на перший погляд, може здатися, що працівники сфери інформаційних технологій не схильні до ризиків на виробництві, та якщо більш глибоко розглянути умови і специфіку праці фахівців сфері ІТ-індустрії, можна виявити ряд факторів які будуть мати негативний вплив на стан охорони праці, та на самого ІТ-фахівця зокрема. Сюди можна віднести як невідповідність освітлення, так і високий рівень шуму, що негативно позначатимуться як на емоційному так і на фізичному стані фахівця, призводитимуть до зниження ефективності праці та виробничих травм. Також, важливим моментом охорони праці ІТ-фахівця є врахування його психологічних можливостей (швидкість реакції, особливості пам'яті та уваги, емоційний стан, тощо). Для того, щоб забезпечити ефективну роботу ІТ-фахівця, потрібно враховувати та максимально компенсувати такі негативні фактори як: надмірне нервово-емоційне навантаження, довготривалі статичні перевантаження, обмежена рухова активність. Всі ці чинники призводить до різноманітних відхилень у стані здоров'я, зокрема до перевтоми, зниження фізичної та розумової працездатності, неврозів, захворювань серцево-судинної системи тощо. Метою даного розділу є огляд конкретних умов праці спеціаліста у сфері ІТ-індустрії. Завданнями для даного розділу є: аналіз умов праці на робочому місці фахівця ІТ-індустрії, розробка конкретних рекомендацій щодо покращення умов

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		128

праці фахівців ІТ-індустрії, огляд пожежної безпеки на ІТ-підприємстві та розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці.

8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;
- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		129

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення їх привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні створювати сприятливі умови для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або професійних захворювань.

[2]

8.3 Пропозиції щодо підвищення працездатності ІТ-фахівців

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів під-

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		131

приємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів.

Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Для більшого розуміння, пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на декілька категорій:

1 Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

2 Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів, які

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		132

пройшли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

8.4 Розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці

Приміщення з ПК повинні мати природне і штучне освітлення, яке відповідало б вимогам ДБН В.2.5-28-2006 «Природне і штучне освітлення» [1], ДСанПІН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2].

Віконні прорізи повинні бути орієнтовані на північ або на північний схід, забезпечувати коефіцієнт природної освітленості (К.П.О.) не менш 1,5% і мати жалюзі або штори. Віконні прорізи повинні мати регульовані пристрої для відкривання, а також жалюзі, завіски, зовнішні козирки тощо. Приміщення із ПЕОМ повинні бути обладнані системою загального рівномірного освітлення. У виробничих і адміністративно-суспільних приміщеннях, де переважно ведеться робота з документами, допускається комбінована система штучного освітлення.

Штучне освітлення має здійснюватися системою загального рівномірного освітлення, яка включає суцільні або такі, що перериваються лінії світильників, розташованих збоку робочих місць (переважно ліворуч), паралельно лінії зору користувачів ПК. Світильники повинні мати розсіювачі світла. У світильниках місцевого освітлення можна використовувати лампи накаливання. При розміщенні ПК по периметру приміщення лінії світильників штучного освітлення

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		133

A – ширина приміщення, $A = 6$ м;

B – довжина приміщення, $B = 6,8$ м.

Підставимо всі значення у формулу та визначимо індекс приміщення:
 $i=1,1$.

Знаючи індекс приміщення, знаходимо $n = 0,46$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп) [7]. Підставимо всі значення у формулу, визначимо світловий потік: $F=43904$ Лм.

Для розрахунку будемо використовувати світлодіодні стельові панелі Lezard 6400K 72 Вт., світловий потік яких $F_{л} = 7200$ Лм.

Число ламп визначається по формулі:

$$N=F/F_{л}$$

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

Підставимо всі значення у формулу та визначимо індекса приміщення

$$N= 43904 / 7200=6,09 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 7 шт.

8.5 Висновки до розділу

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

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		136

9 ОСНОВНІ ВИСНОВКИ

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

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 управління гіперЦОД.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем управління гіперЦОД.
- Досліджена система управління гіперЦОД.
- На основі отриманих результатів досліджень створена програмна реалізація системи управління гіперЦОД.

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

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.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		137

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 RSA.

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

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

					VKPM-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		138

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мороз С.М. Дослідження та програмна реалізація системи управління гіперЦОД // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		139

9. 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.

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

11. 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.

12. 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.

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

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		142

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

31. 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.

32. 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.

33. 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

34. 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.

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

36. 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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		143

37. 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.

38. 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.

39. 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.

40. 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).

41. 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

42. 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.

43. 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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		144

44. 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.

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

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

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

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

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

50. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

51. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		145

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.24.0004.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Мороз С.М.				<i>Дослідження та програмна реалізація системи управління гіперЦОД</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнов С.А.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23Мз			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи управління гіперЦОД.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 21-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи управління гіперЦОД.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи управління гіперЦОД;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинні бути розглянуті пропозиції щодо підвищення працездатності ІТ-фахівців.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 145 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 17.12.2024 р.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Смірнов С.А.

*Дослідження та програмна реалізація
системи управління гіперЦОД*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 17

Літера: РП

Кропивницький – 2024 року

Основна програма

```
# Імпорт модулів для роботи з випадковими числами і
# математичними обчисленнями
import random
import math

# Клас для представлення обчислювальних ресурсів
class ComputeResource:
# Конструктор приймає потужність ресурсу
    def __init__(self, power):
        self.power = power
        self.usage = 0

# Метод для перевірки наявної потужності ресурсу
    def get_available_power(self):
        return self.power - self.usage

# Метод для виділення певної частини ресурсу
    def allocate(self, amount):
# Перевіряємо чи вистачає потужності для виділення
        if self.get_available_power() >= amount:
            self.usage += amount
            return True
        else:
            return False

# Метод для звільнення використаних ресурсів
    def deallocate(self, amount):
# Перевіряємо чи можна звільнити таку кількість ресурсів
        if self.usage - amount >= 0:
            self.usage -= amount
            return True
        else:
            return False

# Клас для представлення сховища даних
class Storage:
# Конструктор приймає об'єм сховища
    def __init__(self, capacity):
        self.capacity = capacity
        self.used = 0

# Метод для перевірки доступного об'єму сховища
    def get_available_storage(self):
        return self.capacity - self.used

# Метод для виділення місця в сховищі
    def allocate_storage(self, size):
# Перевіряємо чи вистачає місця
        if self.get_available_storage() >= size:
            self.used += size
            return True
        else:
            return False

# Метод для звільнення місця в сховищі
    def deallocate_storage(self, size):
# Перевіряємо чи можливо звільнити вказану кількість місця
        if self.used - size >= 0:
            self.used -= size
            return True
        else:
            return False

# Клас для представлення мережевих ресурсів
class NetworkResource:
# Конструктор приймає пропускну здатність мережі
```

```

def __init__(self, bandwidth):
    self.bandwidth = bandwidth
    self.used_bandwidth = 0

# Метод для перевірки доступної пропускної здатності
def get_available_bandwidth(self):
    return self.bandwidth - self.used_bandwidth

# Метод для виділення пропускної здатності мережі
def allocate_bandwidth(self, amount):
# Перевіряємо чи вистачає пропускної здатності
    if self.get_available_bandwidth() >= amount:
        self.used_bandwidth += amount
        return True
    else:
        return False

# Метод для звільнення пропускної здатності
def deallocate_bandwidth(self, amount):
# Перевіряємо чи можливо звільнити таку кількість
    if self.used_bandwidth - amount >= 0:
        self.used_bandwidth -= amount
        return True
    else:
        return False

# Клас управління гіперЦОД
class HyperDatacenterManagement:
# Конструктор приймає параметри ресурсів для управління
    def __init__(self, compute_power, storage_capacity, network_bandwidth):
        self.compute = ComputeResource(compute_power)
        self.storage = Storage(storage_capacity)
        self.network = NetworkResource(network_bandwidth)

# Метод для обробки запитів на ресурси
    def request_resources(self, compute, storage, bandwidth):
# Перевіряємо наявність усіх необхідних ресурсів
    if self.compute.allocate(compute) and
self.storage.allocate_storage(storage) and
self.network.allocate_bandwidth(bandwidth):
        return True
    else:
        return False

# Метод для звільнення використаних ресурсів
    def release_resources(self, compute, storage, bandwidth):
# Звільняємо всі ресурси, які використовувались
    self.compute.deallocate(compute)
    self.storage.deallocate_storage(storage)
    self.network.deallocate_bandwidth(bandwidth)

# Метод для отримання статусу всіх ресурсів
    def get_status(self):
        compute_status = f"Compute Power Available:
{self.compute.get_available_power()} / {self.compute.power}"
        storage_status = f"Storage Available:
{self.storage.get_available_storage()} / {self.storage.capacity}"
        network_status = f"Network Bandwidth Available:
{self.network.get_available_bandwidth()} / {self.network.bandwidth}"
        return compute_status, storage_status, network_status

# Функція моделювання навантаження на ЦОД
def simulate_workload(datacenter, iterations):
    for i in range(iterations):
# Генеруємо випадкове навантаження
        compute_request = random.randint(10, 100)
        storage_request = random.randint(50, 500)
        bandwidth_request = random.randint(10, 50)

```

```

# Виводимо запити на ресурси
    print(f"Iteration {i + 1}: Requesting Compute: {compute_request},
Storage: {storage_request}, Bandwidth: {bandwidth_request}")

# Перевіряємо чи можемо виділити ресурси
    if datacenter.request_resources(compute_request, storage_request,
bandwidth_request):
        print("Resources allocated successfully.")
    else:
        print("Resource allocation failed. Insufficient resources.")

# Виводимо статус після кожної ітерації
    status = datacenter.get_status()
    print(f"Status: {status}")

# Звільняємо ресурси після завершення запиту
    datacenter.release_resources(compute_request, storage_request,
bandwidth_request)

# Основний код для запуску системи управління гіперЦОД
if __name__ == "__main__":
# Створюємо об'єкт гіперЦОД з параметрами
    hyper_dc = HyperDatacenterManagement(compute_power=1000,
storage_capacity=5000, network_bandwidth=1000)

# Моделюємо робоче навантаження з 10 ітераціями
    simulate_workload(hyper_dc, iterations=10)

# Додаємо додаткову функцію для розрахунку ефективності
def calculate_efficiency(datacenter):
# Розраховуємо відсоткове використання ресурсів
    compute_efficiency = (datacenter.compute.usage / datacenter.compute.power) *
100
    storage_efficiency = (datacenter.storage.used / datacenter.storage.capacity)
* 100
    network_efficiency = (datacenter.network.used_bandwidth /
datacenter.network.bandwidth) * 100

# Виводимо результати
    print(f"Compute Efficiency: {compute_efficiency}%")
    print(f"Storage Efficiency: {storage_efficiency}%")
    print(f"Network Efficiency: {network_efficiency}%")

# Викликаємо функцію для обчислення ефективності
    calculate_efficiency(hyper_dc)

```

Файл main.py

```
# main.py
from backup_restore import BackupRestoreManager
from security import SecurityManager
from energy_optimization import EnergyOptimizer
from cloud_integration import CloudIntegrationManager
from maintenance import MaintenanceManager
from reporting import ReportManager
from global_monitoring import GlobalDatacenterManager
from resilience_manager import ResilienceManager
from datacenter_management import HyperDatacenterManagement # Ваш основний клас
управління ЦОД

# Створюємо кілька дата-центрів
dc1 = HyperDatacenterManagement(1000, 5000, 1000)
dc2 = HyperDatacenterManagement(1500, 6000, 1200)
datacenters = {'Region 1': dc1, 'Region 2': dc2}

# Ініціалізуємо модулі
backup_restore = BackupRestoreManager(dc1)
security = SecurityManager()
energy_optimizer = EnergyOptimizer(dc1)
cloud_integration = CloudIntegrationManager(['AWS', 'Azure'])
maintenance = MaintenanceManager(dc1)
reporting = ReportManager(dc1)
global_monitoring = GlobalDatacenterManager(datacenters)
resilience_manager = ResilienceManager(dc1)

# Приклад використання функціоналу

# Резервне копіювання та відновлення
backup_restore.create_backup()
backup_restore.restore_from_backup()

# Управління безпекою
security.add_user("admin_user", "admin")
security.check_access("admin_user", "compute resources")

# Оптимізація енергоспоживання
energy_optimizer.optimize_energy()

# Інтеграція з хмарними провайдерами
cloud_integration.offload_to_cloud("AWS", 100, 200, 50)

# Автоматизація обслуговування
maintenance.schedule_maintenance("2024-10-25 10:00")

# Генерація звітів та аналітика
reporting.generate_report()
reporting.generate_analytics()

# Глобальний моніторинг та управління
global_monitoring.monitor_datacenters()
global_monitoring.manage_resources_globally()

# Адаптація до непередбачуваних подій
resilience_manager.handle_failure()
resilience_manager.reallocate_resources()
```

Файл `resilience_manager.py`

```
# resilience_manager.py

# Клас для управління адаптацією до непередбачуваних подій
class ResilienceManager:
    def __init__(self, datacenter):
        self.datacenter = datacenter

# Метод для реагування на несподівані збої
    def handle_failure(self):
        print("Handling unexpected failure...")
        self.datacenter.auto_scale()
        print("Resources scaled up to handle the issue.")

# Метод для перепланування ресурсів у випадку аварії
    def reallocate_resources(self):
        if self.datacenter.compute.get_available_power() <
self.datacenter.compute.power * 0.2:
            print("Reallocating compute resources due to failure.")
        if self.datacenter.storage.get_available_storage() <
self.datacenter.storage.capacity * 0.2:
            print("Reallocating storage resources due to failure.")
        if self.datacenter.network.get_available_bandwidth() <
self.datacenter.network.bandwidth * 0.2:
            print("Reallocating network resources due to failure.")
```

```
# global_monitoring.py

# Клас для управління кількома ЦОД у різних регіонах
class GlobalDatacenterManager:
    def __init__(self, datacenters):
        self.datacenters = datacenters

# Метод для моніторингу всіх ЦОД
    def monitor_datacenters(self):
        for region, datacenter in self.datacenters.items():
            print(f"Monitoring datacenter in {region}...")
            compute_status, storage_status, network_status =
datacenter.get_status()
            print(f"{region} Compute: {compute_status}")
            print(f"{region} Storage: {storage_status}")
            print(f"{region} Network: {network_status}")

# Метод для глобального управління ресурсами
    def manage_resources_globally(self):
        for region, datacenter in self.datacenters.items():
            print(f"Managing resources in {region}...")
            datacenter.auto_scale()
            compute_status, storage_status, network_status =
datacenter.get_status()
            print(f"{region} Compute after scaling: {compute_status}")
```

Файл reporting.py
Файл X.py

```
# reporting.py

# Клас для генерації звітів та аналітики
class ReportManager:
    def __init__(self, datacenter):
        self.datacenter = datacenter

# Метод для створення звіту про використання ресурсів
    def generate_report(self):
        compute_status, storage_status, network_status =
self.datacenter.get_status()
        print("Generating report...")
        print(f"Compute Report: {compute_status}")
        print(f"Storage Report: {storage_status}")
        print(f"Network Report: {network_status}")
        print("Report generated successfully.")

# Метод для аналітики використання ресурсів
    def generate_analytics(self):
        compute_efficiency = (self.datacenter.compute.usage /
self.datacenter.compute.power) * 100
        storage_efficiency = (self.datacenter.storage.used /
self.datacenter.storage.capacity) * 100
        network_efficiency = (self.datacenter.network.used_bandwidth /
self.datacenter.network.bandwidth) * 100
        print(f"Compute Efficiency: {compute_efficiency}%")
        print(f"Storage Efficiency: {storage_efficiency}%")
        print(f"Network Efficiency: {network_efficiency}%")
```

Файл maintenance.py

```
# maintenance.py

# Клас для автоматизації обслуговування ЦОД
class MaintenanceManager:
    def __init__(self, datacenter):
        self.datacenter = datacenter

# Метод для планового технічного обслуговування
    def schedule_maintenance(self, time):
        print(f"Scheduled maintenance at {time}.")
        self.perform_maintenance()

# Метод для виконання технічного обслуговування
    def perform_maintenance(self):
        print("Performing maintenance: checking hardware, updating software...")
        # Виконання обслуговування
        print("Maintenance completed successfully.")
```

КБПЗ_2024

Файл cloud_integration.py

```
# cloud_integration.py

# Клас для інтеграції з хмарними провайдерами
class CloudIntegrationManager:
    def __init__(self, cloud_providers):
        self.cloud_providers = cloud_providers

# Метод для передачі навантаження на хмарні ресурси
    def offload_to_cloud(self, provider_name, compute, storage, bandwidth):
        if provider_name in self.cloud_providers:
            print(f"Offloading {compute} compute, {storage} storage, and
{bandwidth} bandwidth to {provider_name}.")
        else:
            print(f"Cloud provider {provider_name} not available.")
```

КБПЗ_2024

Файл energy_optimization.py

```
# energy_optimization.py

# Клас для управління енергоспоживанням в ЦОД
class EnergyOptimizer:
    def __init__(self, datacenter):
        self.datacenter = datacenter

# Метод для оптимізації енергоспоживання
def optimize_energy(self):
    if self.datacenter.compute.usage < self.datacenter.compute.power * 0.5:
        print("Optimizing energy by reducing compute power...")
        self.datacenter.compute.power *= 0.9

    if self.datacenter.storage.used < self.datacenter.storage.capacity * 0.5:
        print("Optimizing energy by reducing storage capacity...")
        self.datacenter.storage.capacity *= 0.9

    if self.datacenter.network.used_bandwidth <
self.datacenter.network.bandwidth * 0.5:
        print("Optimizing energy by reducing network bandwidth...")
        self.datacenter.network.bandwidth *= 0.9
```

КБПЗ_2024

Файл security.py

```
# security.py

# Клас для управління безпекою в ЦОД
class SecurityManager:
    def __init__(self):
        self.users = {}
        self.roles = {"admin": [], "user": []}

# Метод для додавання користувача з роллю
def add_user(self, username, role):
    if role in self.roles:
        self.users[username] = role
        print(f"User {username} added with role {role}.")
    else:
        print("Invalid role.")

# Метод для перевірки доступу користувача
def check_access(self, username, resource):
    role = self.users.get(username)
    if role == "admin":
        print(f"Access granted to {resource} for {username}.")
    elif role == "user":
        print(f"Limited access to {resource} for {username}.")
    else:
        print(f"Access denied to {resource} for {username}.")
```

Файл backup_restore.py

```
# backup_restore.py

# Клас для управління резервним копіюванням і відновленням
class BackupRestoreManager:
    def __init__(self, datacenter):
        self.datacenter = datacenter
        self.backup_data = {}

# Метод для створення резервної копії
def create_backup(self):
    self.backup_data = {
        'compute_usage': self.datacenter.compute.usage,
        'storage_used': self.datacenter.storage.used,
        'network_bandwidth_used': self.datacenter.network.used_bandwidth
    }
    print("Backup created successfully.")

# Метод для відновлення даних із резервної копії
def restore_from_backup(self):
    if self.backup_data:
        self.datacenter.compute.usage = self.backup_data['compute_usage']
        self.datacenter.storage.used = self.backup_data['storage_used']
        self.datacenter.network.used_bandwidth = self.backup_data['network_bandwidth_used']
        print("Restored from backup successfully.")
    else:
        print("No backup available.")
```

Файл `virtualization_support.py`

```
# virtualization_support.py

# Клас для управління підтримкою віртуалізації у ЦОД
class VirtualizationManager:
    def __init__(self, datacenter):
        self.datacenter = datacenter

# Метод для підтримки різних типів віртуалізації
    def allocate_virtual_resources(self, vm_resources, container_resources):
        vm_allocation = self.datacenter.request_resources(vm_resources[0],
vm_resources[1], vm_resources[2])
        container_allocation =
self.datacenter.request_resources(container_resources[0],
container_resources[1], container_resources[2])
        if vm_allocation:
            print("Resources allocated for virtual machines.")
        if container_allocation:
print("Resources allocated for containers.")
        if not vm_allocation or not container_allocation:
print("Failed to allocate resources for virtual environments.")
```

КБПЗ_2024

Файл load_balancer.py

```
# load_balancer.py

# Клас для балансування навантаження між двома дата-центрами
class LoadBalancer:
    def __init__(self, datacenter1, datacenter2):
        self.datacenter1 = datacenter1
        self.datacenter2 = datacenter2

# Метод для розподілу навантаження між дата-центрами
def balance(self, compute, storage, bandwidth):
    if not self.datacenter1.request_resources(compute, storage, bandwidth):
        print("Redirecting load to another data center...")
        if not self.datacenter2.request_resources(compute, storage,
bandwidth):
            print("Both data centers are overloaded.")
        else:
            print("Resources allocated at another data center.")
    else:
        print("Resources allocated successfully at primary data center.")
```

КБПЗ_2024

Файл prediction.py

```
# prediction.py

# Клас для прогнозування навантаження на основі історії використання ресурсів
class LoadPredictor:
    def __init__(self, history):
        self.history = history

# Метод для прогнозування майбутнього навантаження
    def predict(self):
        avg_compute = sum([h[0] for h in self.history]) / len(self.history)
        avg_storage = sum([h[1] for h in self.history]) / len(self.history)
        avg_bandwidth = sum([h[2] for h in self.history]) / len(self.history)
        return avg_compute, avg_storage, avg_bandwidth

# Виведення прогнозу
    def display_prediction(self):
        compute, storage, bandwidth = self.predict()
        print(f"Predicted future load: Compute: {compute}, Storage: {storage},
        Bandwidth: {bandwidth}")
```

КБПЗ_2024

Файл autoscaling.py

```
# autoscaling.py

# Клас для автоматичного масштабування ресурсів у ЦОД
class AutoScaler:
    def __init__(self, datacenter):
        self.datacenter = datacenter

# Метод для автоматичного масштабування ресурсів
    def auto_scale(self):
        if self.datacenter.compute.get_available_power() <
self.datacenter.compute.power * 0.2:
            print("Scaling compute resources...")
            self.datacenter.compute.power *= 1.5
        if self.datacenter.storage.get_available_storage() <
self.datacenter.storage.capacity * 0.2:
            print("Scaling storage resources...")
            self.datacenter.storage.capacity *= 1.5
        if self.datacenter.network.get_available_bandwidth() <
self.datacenter.network.bandwidth * 0.2:
            print("Scaling network resources...")
            self.datacenter.network.bandwidth *= 1.5
```

КБПЗ_2024