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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти

на тему

“Програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі”

КбПЗ_2026

Виконав здобувач вищої освіти
IV курсу, групи КІ-23мбз
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Богаш О.О.
«____» _____ 2026 р.

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

м. Кропивницький

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Освітній ступінь *бакалавр*

Галузь знань _12 _____*"Інформаційні технології"*_____

Спеціальність 123 *"Комп'ютерна інженерія"*

Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

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

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

Богаш Оксані Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі*

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

затверджені наказом вищого навчального закладу № 131-02 від 27.02.2026 року

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

4. Мета та завдання випускної кваліфікаційної роботи*: Метою роботи є розробка програмного забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі*

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

1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію.
6. Висновки

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

Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

КОПІЯ 2026

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

КАЛЕНДАРНИЙ ПЛАН

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

КбПЗ_2026

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника
_____ Смірнов О.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача
_____ Богаш О.О.
(прізвище та ініціали)

АНОТАЦІЯ

Богаш О.О. Програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

Метою розробки є програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

Результат роботи – програмна реалізація системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

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, Composable Infrastructure System

КПЗ – 2026

ABSTRACT

Bogash O.O. Software for the Composable Infrastructure System for dynamic allocation of computer network resources. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for the Composable Infrastructure System for dynamic allocation of computer network resources.

The purpose of the development is software for the Composable Infrastructure System for dynamic allocation of computer network resources.

The result of the work is a software implementation of the Composable Infrastructure System for dynamic allocation of computer network resources.

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

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

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

The program is developed in Python.

Keywords: computer engineering, Composable Infrastructure System

КБПЗ_2026

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 2
ВСТУП..... 3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 5
1.1 Призначення системи..... 5
1.2 Область застосування..... 6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти..... 10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 29
2.3 Розгорнута постановка завдання 30
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 31
3.1 Опис функціонування системи 31
3.2 Розробка структурної схеми..... 36
3.3 Розробка функціональної схеми 39
3.4 Розробка діаграми процесів..... 53
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 55
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 55
4.2 Захист розробленого програмного забезпечення..... 68
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 70
6 ОСНОВНІ ВИСНОВКИ..... 76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 78

КОПІЯ 2026

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 CoSntrol Message Protocol – міжмережевий 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 ймовірність відкидання пакета

КОПЗ_2026

ВСТУП

Актуальність теми. Для задоволення запитів бізнесу з гнучкого надання IT-ресурсів багато компаній роблять вибір на користь конвергентної й гіперконвергентної інфраструктури. Ці рішення дозволяють спростити експлуатацію й підвищити ефективність її використання. Тим часом на ринку набирає популярність нова парадигма – компонуєма інфраструктура.

Компонуємі системи (Composable Infrastructure System, CIS) розглядаються як наступний етап еволюції конвергентних інфраструктур. Основна ідея проста: для виконання конкретного робочого навантаження необхідна конфігурація формується «на льоту» за допомогою програмного забезпечення з наявного пула апаратних ресурсів.

Стимулом до реалізації даного підходу послужило, зокрема, прагнення підприємств одержати в себе на площадці ті ж можливості, що й у хмарі. Як і в хмарній моделі «інфраструктура як сервіс», необхідні ресурси запитуються із загального пула, що розміщається не в публічній хмарі, а в ЦОД клієнта й контролюється їм же.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

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

– Огляд існуючих систем Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– Дослідження системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– Програмна реалізація системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

КОПЗ – 2026

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ_2026

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Існують різні думки про те, якими характеристиками повинна володіти компонентна інфраструктура, оскільки концепція досить нова (хоча її джерела можна звести до мейнфреймам). Так, наприклад, HPE, один з піонерів у даній області, виділяє три обов'язкові характеристики: гнучкий пул ресурсів, уніфікований інтерфейс API і програмно-визначаємий інтелект.

Щоб скласти необхідну конфігурацію відповідно до (тих, що змінюються) вимог застосунків, пул ресурсів з обчислювальних потужностей, засобів зберігання й мережевих компонентів повинен бути досить гнучкий. На його основі програмне забезпечення створює абстрактну фізичну систему, що і надається зовнішньому застосунку.

Щоб такий підхід працювали, компонуємі системи повинні бути не відрізняємими від типових фізичних систем – з погляду програмного забезпечення, що на них буде виконуватися. Крім того, компонування й повернення ресурсів, що звільнилися, у пул варто організувати таким чином, щоб це не впливало на роботу інших систем.

Для динамічного розподілу ресурсів необхідний інтелект. Засоби компонування й оркестрації ресурсів дозволяють різним додаткам і пристроям працювати зкоординовано. А повнофункціональна система керування забезпечує автоматизоване виявлення ресурсів у пулі й полегшує реалізацію політики їхнього надання.

Керування компонуємими системами здійснюється через API. Як вважають в Forrester, наявність опублікованого набору API, за допомогою якого користувачі й застосунки могли б одержати доступ до керування компонуємою інфраструктурою, є обов'язковим, тоді як підтримка графічного інтерфейсу й

КПЗ_2026

інтерфейсу командного рядка опціональні. В HPE підкреслюють важливість уніфікованих API для реалізації уніфікованого інтерфейсу керування.

В Forrester розрізняють локальну й глобальну компонуємі інфраструктури. У локальному варіанті вона забезпечує детальний вибір устаткування аж до рівня окремих процесорів і модулів пам'яті. У глобальному передбачаються віртуалізовані конструкти – наприклад, контейнери й віртуальні машини, які користувачі можуть задіяти в межах декількох площадок.

Для з'єднання різних компонентів потрібна високошвидкісна мережа з низькою затримкою. Таким чином, ключовим фактором (локальної) компонуємої інфраструктури в довгостроковій перспективі стає заміна традиційних мідних з'єднань на оптичні. Однак, як відзначають в 451 Research, оптична фотоника для організації з'єднань між стійками навряд чи стане широко доступна протягом найближчих трьох-п'яти років. Відповідно, наявні на ринку системи поки обмежуються переважно масштабами стійки.

1.2 Область застосування

Як вказується в дослідженні «Дорога до компонуємої інфраструктури», що провела аналітична компанія Moor Insights & Strategy, першими кандидатами на впровадження даної архітектури є організації, що використовують гіперконвергентну інфраструктуру, оскільки вони вже знайомі з перевагами єдиної платформи, що поєднує обчислювальні потужності, ємності зберігання й мережеві ресурси. Але навіщо їм це? Чим відрізняється компонуєма інфраструктура від доступних повсюдно рішень, таких, наприклад, як гіперконвергентна інфраструктура?

Список її переваг багато в чому перетинається з аналогічними переліками для HCI: підвищення ступеня використання ресурсів, прискорення розгортання застосунків, скорочення капітальних і операційних витрат і т.д. і т.п. Правда, як

КПЗ_2026

затверджується, у цьому випадку вигода виявляється ще більшої, ніж у попередників.

Кінцевий обсяг доступного місця обмежує волю маневру при компонуванні стійкових серверів, де необхідно розмістити процесори, пам'ять, диски, шлейфи, блоки живлення, вентилятори й т.д. Це означає, що конфігурація типового сервера звичайно не оптимальна з погляду виконуваного навантаження: застосунку з високими обчислювальними навантаженнями не задіють повною мірою пам'ять і диски, а у випадку застосунків з високими навантаженнями вводу-виводу, навпаки, процесори й пам'ять виявляються надлишковими.

Для досягнення найбільшої ефективності замовники намагаються підібрати конфігурацію відповідно до планованого навантаження. Однак, керування й обслуговування множини різних конфігурацій приводять до ускладнення інфраструктури, а в остаточному підсумку – до розуміння того, що необхідно створити нову серверну архітектуру – загальну платформу, на якій могли б виконуватися різні застосунки.

Завдяки дезагрегованій архітектурі компонуєма інфраструктура дозволяє використовувати ресурси більш ефективно, ніж гіперконвергентні рішення. За деякими оцінками, в існуючих гіпермасштабованих центрах обробки даних утилізація ресурсів становить усього 45% (у середньому за добу). Як затверджують в Western Digital, компонуєма архітектура дозволяє довести цей показник до 70% і вище.

Оптимальна конфігурація системи й більше повне використання ресурсів дозволяють скоротити інвестиції в необхідне устаткування й здешевить надання сервісів на базі компонуємої архітектури. По оцінці WD, при використанні даного підходу фінансових вкладень буде потрібно на 50% менше, а вартість володіння скоротиться на 40% (у порівнянні із традиційними гіпермасштабованими рішеннями).

Підприємства нерідко мають одну інфраструктуру для успадкованих застосунків, а іншу – для мобільних і хмарних. Застосування компонуємої

КОПЗ_2026

архітектури дозволяє ліквідувати розрив між старими системами й новими хмарними додатками: на тому самому устаткуванні можна розгортати й ті й інші. Це дозволяє одночасно підтримувати різні класи навантажень – традиційні серверні (bare metal в англійській термінології, тобто запускаються на фізичному сервері), віртуалізовані й контейнеризовані – більш ефективно, ніж існуючі технології віртуалізації.

Таким чином, компонуєма інфраструктура виявляється привабливою альтернативою переміщенню робочих навантажень у публічні хмари, надаючи при цьому необхідну гнучкість для підтримки мінливих навантажень. Це відповідає бажанню багатьох підприємств залишити ключові системи на власній площадці під своїм контролем.

Недоліки компонуємої архітектури

Компонуєма архітектура припускає повну дезагрегацію різних серверних компонентів, але поки це технічно неможливо. Для рознесення процесорів і пам'яті потрібні мережі з дуже високою пропускною здатністю й зверхмалою затримкою, однак необхідна для цього кремнієва фотоника поки перебуває в стадії розробки.

IT-інфраструктура усе ще недостатньо гнучка, щоб її компоненти можна було комбінувати довільним образом. В ідеалі компонуєма архітектура повинна дозволяти поєднувати процесорні сокети різних серверів, щоб з них можна було сформувати один логічний процесор.

Такі ресурси, як прискорювачі (GPU, FPGA, ASIC), пам'ять для тривалого зберігання даних (Storage Class Memory, SCM), мережеві інтерфейси, бажано комбінувати «на льоту» довільним образом відповідно до запитів застосунків. Однак це неможливо, поки пам'ять прив'язана до процесора усередині сервера. Необхідно, щоб серверна архітектура будувалася навколо пам'яті, а не процесорів (memory centric).

Проте, завдяки новітнім технологіям, зокрема загальним інтерфейсам прикладного програмування (API), функціональність «інфраструктура як код»

КОПЗ_2026

може бути реалізована вже сьогодні, однак її потенціал обмежений можливостями дезагрегації нижчележачого устаткування. Поки ж компонуєма (програмно-визначаєма) і дезагрегируема (апаратна) архітектури розвиваються різними шляхами.

Пул ресурсів для компонуємої інфраструктури повинен охоплювати все наявне устаткування в центрі обробки даних. Однак IT-інфраструктура сучасного ЦОД формується з різнотипного устаткування множини різних вендорів. На сьогоднішній день немає стандартизованих способів для оркестрації такої інфраструктури.

У результаті при реалізації компонуємої архітектури замовники виявляються обмежені у виборі серверами й іншим устаткуванням конкретного виробника. Таким чином, компонуєма інфраструктура, підвищуючи гнучкість масштабування обчислювальних ресурсів, породжує ту ж залежність від одного вендора, як і гіперконвергентна.

Одним з потенційних переваг компонуємої архітектури є спрощення адміністрування й керування. Дійсно, використання готових сценаріїв і шаблонів для формування необхідної інфраструктури з набору компонентів спрощує обслуговування IT, але для створення цих шаблонів і адаптації коду потрібні фахівці більше високої кваліфікації.

Таким чином, компонуємій архітектурі треба буде пройти ще довгий шлях розвитку. Поки ж набір компонентів, які можна комбінувати довільним образом, обмежений процесорами, жорсткими дисками й флеш-накопичувачами.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КПЗ-2026

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

HPE Synergy: від блейд-платформи до компонуємої інфраструктури

У 2025-му компонуєма платформа HPE Synergy, випущена п’ять років тому, не тільки зберегла, але й зміцнила свої позиції. Про це говорить те, що найближчий конкурент HPE – компанія Dell – на конференції Dell Technologies World анонсувала випуск системи зі схожими функціональними можливостями.

HPE Synergy повною мірою реалізувала концепцію «інфраструктура як код» (infrastructure-as-code). Платформа оптимальна для швидкого розгортання традиційних і хмарних застосунків і запуску критично важливих бізнес-процесів у компонуємих програмно-визначаємих ЦОДах. За результатами незалежних досліджень вона відповідає критеріям системи компонуємої інфраструктури CIS (composable infrastructure system) і дотепер уважається новаторською.

У свій час HPE Synergy виросла із блейд-системи HPE BladeSystem C7000, що займала 10 юнитов у стійці й дозволяла розмістити 16 «лез» і/або модулів зберігання даних плюс додаткові мережеві компоненти. На відміну від попередниці, HPE Synergy – це не просто блейд-система, але й програмно-визначаєма система зберігання з усіма перевагами конвергентних і гіперконвергентних архітектур. Останній актуальний продукт HPE – Synergy 12000 з 12 слотами для установки «лез» і по одному слоту для Composer’а й Image Streame’ра. Перший використовується для керування засобами OneView, другий – для зберігання образів операційних систем.

Модуль HPE Synergy

Успіх HPE Synergy підтверджують реальні цифри. До середини 2025 року на HPE Synergy перейшло 1400+ замовників по усьому світі. Використовуючи

КОПЗ_2026

єдиний користувальницький інтерфейс, вони управляють великими ЦОДами й оптимізують цей продукт під параметри конкретної інфраструктури за допомогою єдиного програмувального API. Тепер замість рутинних ручних процедур потрібно написати всього кілька рядків коду. Більшу роль у поширенні HPE Synergy зіграла проста інтеграція з популярними додатками для автоматизації – Chef, Puppet, Red Hat, VMware, Microsoft, Docker і іншими. За допомогою програмно-визначаємих шаблонів IT-фахівець може швидко й гарантовано розгорнути необхідну інфраструктуру для будь-яких застосунків, використовуючи мінімум кабелів і пасивного мережевого устаткування.

Кому підійде HPE Synergy

HPE Synergy добре зарекомендує себе в умовах статичної й динамічної інфраструктури, у тому числі там, де досить 1-2 блейд-кошиків, а не 10 або 20. Але максимальна економічна ефективність досягається в системах, де шасі заповнене на 80-100 %.

Платформа буде актуальна й там, де не передбачається використання зовнішніх СЗД. У цьому випадку диски з дискового модуля в Synergy підключаються прямо до обраних серверів в ексклюзивне користування. З іншого боку, платформа підтримує бездискові леза. Але якщо накопичувачі все-таки плануються, то це можуть бути як NVMe, так і SAS, причому допускається їхнє спільне використання.

Здатність витримувати високі робочі навантаження – ще одна сильна сторона HPE Synergy. Вона впорається з установкою ОС прямо на сервер, розгортанням гіпервізора або контейнера. Паралельно Synergy вирішить ще одну проблему: зменшить споживання енергії й скоротить виділення тепла в навколишнє середовище. Відповідно, бізнес буде витрачати менше ресурсів на оплату електроенергії й кондиціювання приміщень.

Виробник рекомендує використовувати HPE Synergy для побудови суперкомп'ютерів кластерного типу, організації хмар, розгортання систем віртуалізації, аналітичних розрахунків в Big Data, а також для підтримки мобільних сервісів. Вона успішно вирішує завдання бізнесу зі штатом від 200

КПІ 2026

співробітників і великих ЦОДів. Їхні клієнти завдяки цій платформі заощаджують на орендній платі, оптимізують процес закупівлі нового обладнання й одержують більше гнучкості в конфігуруванні системи.

С Synergy бізнес буде стабільний не тільки сьогодні-завтра, але й у довгостроковій перспективі. Платформа вже відкрита до інновацій: HPE недавно продемонстрували прототип комп'ютера Gen-Z з універсальною пам'яттю й шиною нового покоління. Обидві ці технології підтримуються Synergy.

Red Hat OpenStack Platform 15

Компанія Red Hat випустила Red Hat OpenStack Platform 15, чергову версію високо масштабованої й динамічної хмарної IaaS (as-a-Service). Ця версія базується на релізі Stein відкритого проекту OpenStack і пропонує більше продуктивності й безпеці при роботі в хмарі, а також розширену підтримку апаратного забезпечення для більше оперативної й безпечної підтримки вимогливих робочих навантажень. Рішення побудоване на основі операційної системи Red Hat Enterprise Linux 8.

В епоху цифрової трансформації операційна ефективність і здатність динамічно розгортати IT-системи стають ключовим фактором успіху. Дана версія Red Hat OpenStack Platform пропонує досягти цих цілей, об'єднавши інновації співтовариства OpenStack.

Ключові особливості Red Hat OpenStack Platform 15

– Потужна платформа для навантажень, таких як машинне навчання й штучний інтелект, за рахунок підтримки технології Nvidia GPU/vGPU при прийнятті рішень у режимі близькому до реального часу.

– Посилена безпека без зниження продуктивності за рахунок переносу IPSec-шифрування на рівень мережевого адаптера підвищує масштабованість і пропускну здатність, доступну для виробничих навантажень.

– Поліпшена мережа за рахунок дороблено з урахуванням особливостей сучасних мережевих застосунків стека TCP/IP допомагає підвищити аптайм застосунків і безпеку навантажень.

КПІ – 2026

– Розширений вибір апаратних архітектур за рахунок підтримки IBM Power9 LE PowerVM забезпечує додаткові варіанти при побудові відказостійких інфраструктур.

– Підтримка апаратних модулів безпеки HSM для поліпшеної генерації й керування ключами шифрування при використанні підтримуваних HSM-рішень забезпечує більше безпечний і централізований криптографічний обмін повідомленнями.

RHEL 8 підтримується в якості гостьовий ОС на платформі Red Hat OpenStack Platform 13, що має довгий життєвий цикл « long-life release».

Власники діючої підписки Red Hat OpenStack Platform можуть одержати доступ до OpenStack Platform 15 на порталі Red Hat Customer Portal.

Red Hat OpenStack Platform 14

14 листопада 2025 року компанія Red Hat анонсувала Red Hat OpenStack Platform 14, версію свого маштабуємого хмарного IaaS-рішення, повністю готового до розгортання застосунків. Рішення базується на СПО-проекті OpenStack версії Rocky і пропонує більше тісну інтеграцію з Red Hat OpenShift Container Platform, Kubernetes-платформою корпоративного класу, підсилюючи потенціал технологій Kubernetes на платформі OpenStack у корпоративному секторі. Підвищуючи ефективність використання ресурсів «голого заліза» (baremetal серверів) і розширюючи можливості автоматизованого розгортання, Red Hat OpenStack Platform 14 надає єдиний інфраструктурний фундамент для традиційних віртуалізованих навантажень і сучасних хмарних застосунків.

Базуючись на провідному Linux-Дистрибутиві корпоративного класу, Red Hat OpenStack Platform, зі слів розроблювача, дає організації можливість перетворювати свою IT-інфраструктуру в більше гнучку, ефективну й інноваційну IT-середовище. Модульність платформи Red Hat допомагає оптимізувати IT-операції для вже наявних традиційних програмних систем і пропонує надійний фундамент для розробки й розгортання хмарних застосунків нового покоління.

КОПЗ_2026

В Red Hat підкреслили, що в міру того, як перехід на хмарні застосунки й контейнери стає важливим елементом стратегії цифрової трансформації, на перший план виходить можливість ефективно розгортати й масштабувати корпоративні Kubernetes-рішення на платформі OpenStack. Тому Red Hat OpenStack Platform 14 пропонує не тільки хостинг Red Hat OpenShift Container Platform, але й автоматизацію ключових завдань по підготовці й масштабуванню IT-ресурсів для використання Kubernetes-платформи корпоративного класу:

– Автоматизована підготовка ресурсів «голого заліза» і віртуальних інфраструктур для Red Hat OpenShift Container Platform на вузлах Red Hat Enterprise Linux – дозволяє використовувати те саме хмарне рішення як для контейнерних, так і для віртуалізованих робочих навантажень у тих випадках, коли організація рухається убік хмарних навантажень на «голому залізі».

– Автоматизоване розгортання продакшн-кластерів Red Hat OpenShift Container Platform високої готовності (HA) – допомагає спростити шлях до безперервності IT-операцій за рахунок усунення єдиної крапки відмови.

– Інтегрована мережа (Kuryr) – дозволяє контейнерним навантаженням OpenShift і віртуальним навантаженням OpenStack, що належать одному орендареві, використовувати ту саму віртуальну мережу для збільшення продуктивності на рівні архітектури.

– Автоматизоване використання убудованих сервісів балансування навантаження OpenStack для контейнерних front-end навантажень.

– Використання убудованого об'єктного сховища OpenStack для ефективного хостингу контейнерних реєстрів.

В Red Hat OpenStack Platform 14 з'явилася можливість масштабувати вузли Red Hat OpenShift нагору-долілиць за допомогою інструментарію керування Director, що дозволяє динамічно перерозподіляти ресурси в міру зміни вимог робочих навантажень. Це може допомогти підвищити ефективність використання обчислювальних ресурсів, особливо коли організація шукає нові напрямки для бізнесу або вивчає можливості запуску нових послуг, вважають в Red Hat.

КОПЗ_2026

Виконання робочих навантажень нового покоління на «голому залізі» дозволяє максимально повно задіяти потужні обчислювальні можливості сучасного устаткування при переході на хмарні технології. Але тільки за умови єдності й погодженості керування серверами на «голому залізі» і віртуалізованих вузлів, як це зроблено в рішенні Red Hat.

Як відзначив розроблювач, Red Hat OpenStack Platform на «голому залізі» дозволяє використовувати цей клас систем для виконання хмарних навантажень, на зразок тих, що працюють на Red Hat OpenShift Container Platform, з набагато кращою керованістю й погодженістю, чим раніше, більше швидкості й гнучкості без втрати керованості й стабільності. Зокрема, інтеграція Ansible для мережевих пристроїв і механізмів підготовки IT-ресурсів автоматизує реконфігурацію мережевих комутаторів для надійної ізоляції хмарних орендарів. Red Hat OpenStack Platform на «голому залізі» у сполученні з відповідною програмою сертифікації устаткування спрощує й здершевляє вибір апаратних компонентів, гарантовано працюючих з рішенням Red Hat.

Завдяки розширеній інтеграції з Red Hat Ansible Automation, процес розгортання Red Hat OpenStack Platform 14 став простіше, ніж у попередніх версіях. Перед тим, як запускати Red Hat OpenStack у продакшн-режимі, фахівці з експлуатації можуть улаштувати попередній прогін, щоб ідентифікувати й усунути проблеми. Став прозоріше й сам процес розгортання, що також прискорює локалізацію й усунення неполадок, включаючи, при необхідності, повтор кроків по резервуванню обчислювальних потужностей і розгортанню ізольованих середовищ, затверджує розроблювач.

Зі слів представників розроблювача, представлена версія IaaS-рішення Red Hat також пропонує ряд інших функцій, таких як:

– Процесорне масштабування для ресурсномістких навантажень і нових класів завдань, на зразок графічного рендерингу й систем штучного інтелекту за рахунок підтримки як Технологічного прев'ю технології NVIDIA GRID Virtual PC

КОПІЯ 2026

(vPC), що забезпечує спільне використання GPU NVIDIA декількома віртуальними машинами й додаткам.

– Поліпшена робота зі сховищами даних у плані доступності, керованості, безпеці й міграції даних за рахунок додаткової інтеграції з Red Hat Ceph Storage, включаючи можливість спільного використання того самого сховища Cinder декількома віртуальними машинами.

– Включення Skydive, інструмента мережевого аналізу, що спрощує валідацію, документування й усунення неполадок у складних мережевих топологіях (Skydive, версія Tech Preview).

IT-середовище кожної організації унікальні в плані використовуваного устаткування, інфраструктурного ПЗ й клієнтських застосунків. Тому Red Hat прагне запропонувати замовникам Red Hat OpenStack максимально широкий вибір, включаючи не тільки платформи x86, але й у системі IBM Power.

З листопада 2025 року Red Hat OpenStack Platform 14 стане доступна у вигляді самостійного продукту на порталі Red Hat Customer, а також у складі рішень Red Hat Cloud Infrastructure і Red Hat Cloud Suite.

Red Hat OpenStack Platform 13

24 травня 2025 року компанія Red Hat анонсувала Red Hat OpenStack Platform 13, чергову версію свого хмарного IaaS-рішення. Базуючись на СПО-проекті OpenStack версії «Queens», рішення Red Hat пропонує корпоративним замовникам основу для побудови гібридної хмари й проведення цифрової трансформації, мінімізуючи складності за рахунок режиму відновлення Fast Forward і забезпечуючи високу інтеграцію по всьому стеку технологій.

Red Hat OpenStack Platform прискорює доставку застосунків і послуг за рахунок надійної й розширюваної приватної хмари. Рішення дозволяє перетворити корпоративну IT-інфраструктуру в більше гнучке й ефективне середовище, необхідну для досягнення успіху в епоху цифровізації, вважають в Red Hat.

КОПІЯ_2026

За інформацією, наданої розроблювачем, Red Hat OpenStack Platform 13 дозволяє замовникові реалізувати у своєму дата-центрі хмарну архітектуру й домогтися еластичності ресурсів, масштабованості й росту ефективності, у тому числі, за рахунок розширеної інтеграції із хмарними технологіями Red Hat, такими як:

– Red Hat CloudForms – рішення по керуванню гібридною хмарною інфраструктурою;

– Red Hat Ceph Storage – масштабоване інтегроване сховище для швидкого розгортання сотень віртуальних машин на базі одного образа й одержання повністю підтримуваного рішення по зберіганню даних;

– Red Hat OpenShift Container Platform – перетворює Red Hat OpenStack Platform 13 у розширювану платформу для cloud-native робочих навантажень, додаючи до масштабованої й універсальної архітектури Openstack підтримку Linux-Контейнерів і оркестрацію за допомогою Kubernetes.

Щоб спростити відновлення між версіями з довгостроковою підтримкою, Red Hat OpenStack 13 пропонує механізм Fast Forward, що допомагає IT-фахівцям замовника вводити додатковий функціонал у своє середовище OpenStack при мінімальних операційних ризиках. Надаваний через Red Hat OpenStack Platform Director механізм Fast Forward дозволяє вибрати оптимальний режим відновлення: або за прискореною схемою з відновленням раз у півроку й використанням функцій з галузі проекту, підтримуваної силами співтовариства OpenStack; або традиційний варіант із фіксацією на версіях з довгостроковою підтримкою. Додатковий функціонал полегшує перехід з Red Hat OpenStack Platform 10 на версію 13, дозволяючи здійснити його з меншою кількістю перезавантажень і без використання додаткового устаткування, відзначили в Red Hat.

Відповідно до заяви розроблювача, Red Hat OpenStack Platform 13 – хмарна платформа, що сполучить високу стабільність, доступність і безпеку із гнучкими інноваційними можливостями:

КБПЗ-2026

– Red Hat OpenStack Platform 13 розширює охват і поширює контейнеризацію на всі сервіси OpenStack, включаючи мережеві сервіси й служби сховища даних.

– Розширення можливостей OpenStack – інтеграція до складу Red Hat OpenStack Platform 13 відкритих проектів безпеки, таких як OpenStack Barbican, забезпечує керування життєвим циклом критичних даних, включаючи паролі, сертифікати безпеки й ключі, на рівні орендаря (тенанта). Barbican відкриває можливості для використання шифрування, такі як шифровані томи Cinder, підпис образів Glance і шифрування об'єктів Swift. Red Hat OpenStack Platform 13 також розширює охват TLS на внутрішні комунікації для таких сервісів, як VNC, OpenDaylight і Redis, що полегшує дотримання норм безпеки FedRAMP, SecNumCloud і інших галузевих стандартів керування ризиками.

Як відзначив розроблювач платформи, Red Hat OpenStack Platform відрізняється великої екосистемою партнерських рішень, що включає корпоративні продукти Dell EMC, Intel, Lenovo, NetApp і Rackspace, а також користується підтримкою представників телекомунікаційної галузі, включаючи Cisco, Ericsson, Huawei, NEC і Nokia. Завдяки цьому Red Hat OpenStack Platform 13 підходить для розробки, розгортання й підтримки ключових телеком-сервісів і клієнтських застосунків у сфері комунікацій, пропонуючи операторам наступні переваги:

– Інтеграція й підтримка OpenDaylight, модульної відкритої платформи конфігурування й автоматизації програмно-визначаємих мереж, з використанням Red Hat OpenStack Platform Director. Це дозволяє реалізувати переваги адаптивних мережевих рішень на основі ПЗ з відкритим кодом.

– Гіпервізор «KVM реального часу» на основі Red Hat Enterprise Linux Real Time Kernel для середовищ, де потрібні ультра-низькі затримки.

Red Hat OpenStack Platform 13 з'явиться в червні 2025 року як самостійний продукт, а також у складі пакетів Red Hat Cloud Infrastructure і Red Hat Cloud Suite.

КБПЗ-2026

Red Hat OpenStack Platform 12

Red Hat OpenStack Platform 12 – чергова версія масштабованої й гнучкої хмарної платформи класу IaaS. Рішення побудоване на основі відритого проекту OpenStack версії Pike і вперше реалізує контейнерозовані сервіси, що допомагає прискорити розробку застосунків за рахунок більшої гнучкості й меншої складності IT-систем. Red Hat OpenStack Platform 12 також одержала ряд серйозних поліпшень, включаючи модернізовані механізми DCI (distributed continuous integration) і оптимізовані засоби безпеки для керування ризиками й забезпечення відповідності правилам по роботі з даними, говориться в повідомленні Red Hat.

У цілому Red Hat OpenStack Platform 12, побудована на основі ОС Red Hat Enterprise Linux і призначена для створення приватних і загальнодоступних хмарних інфраструктур, являє собою протестовану, сертифіковану й забезпечену підтримкою корпоративного рівня версію СПО OpenStack. До складу Red Hat OpenStack Platform також входить Red Hat CloudForms – мультихмарна платформа керування, що реалізує прозорий операційний контроль і керування інфраструктурою й робочими навантаженнями Red Hat OpenStack Platform на основі політик. Крім того, Red Hat OpenStack Platform 12 тісно інтегрується з Red Hat Ceph Storage, программной СЗД, що забезпечує блокові, файлове й об'єктні сховища даних з горизонтальним масштабуванням.

Контейнеризація сервісів OpenStack

Для прискорення виводу на ринок пропозицій сучасної організації потрібна хмарна інфраструктура, здатна розподіляти ресурси швидше й з більшою масштабованістю. Упакування сервісів OpenStack в Linux-Контейнери вирішують це завдання, підвищуючи гнучкість таких процесів, як модернізація, відкіт до попередньої версії й обслуговування сервісів, і одночасно полегшуючи експлуатацію хмари. Крім того, контейнеризація спрощує масштабування сервісів OpenStack, допомагаючи оперативно реагувати на зміну навантаження по сторони користувачів.

КОПІЯ 2026

Поза залежністю від варіанта розгортання, будь те установка з нуля або автоматизоване відновлення за допомогою інструмента director, Red Hat OpenStack Platform 12 упаковує в контейнери більшість сервісів OpenStack, а також на умовах Technology Preview виконує контейнеризацію мережевих служб і служб зберігання даних. Як результат, стратегічні партнери Red Hat одержують можливість сертифікувати свої драйвери й підключаються модули, що, для цієї моделі розгортання, мінімально торкаючись обслуговування замовників.

Оптимізована безпека

Функції Red Hat OpenStack Platform 12, такі як сервіс automated infrastructure enrollment, допомагають підвищити безпека за рахунок автоматизації керування життєвим циклом сертифікатів безпеки. Компоненти платформи OpenStack Block Storage (Cinder) і Bare Metal Provisioning (Ironic) оновлені для підтримки шифрування томів і поліпшеної роботи з дисковими розділами відповідно. Розширюючи підтримку глобальних ініціатив в області керування ризиками, Red Hat пропонує замовникам Red Hat OpenStack Platform можливість завантажити на порталі Customer Portal посібник з безпеки, у якому розглядаються засоби, способи й рекомендації з реалізації мер безпеки при побудові захищених середовищ OpenStack.

Підвищена гнучкість

Поняття компонуємості вперше з'явилося в Red Hat OpenStack Platform 10 із введенням складових (composable) ролей, наборів з окремих процесів і сервісів, які оператори можуть формувати за своїм розсудом. Red Hat OpenStack Platform 11 розширила використання складених ролей, щоб з їхньою допомогою спростити розгортання й відновлення платформи. Red Hat OpenStack Platform 12 іде ще далі й представляє компонуємі мережі composable networks.

У попередніх версіях Red Hat OpenStack Platform користувачам пропонувався вибір з жорстко заданого набору мережевих топологій, а у версії 12 є можливість більш вільно визначити топологію мережі за допомогою компонуємих мереж. Крім того, оператори можуть визначати необмежену

КПІ – 2026

кількість мереж, у тому числі з використанням популярної топології L3 spine-and-leaf, що спрощує настроювання корпоративних рішень OpenStack з урахуванням специфіки масштабування. Крім того, Red Hat OpenStack 12 тепер підтримує відкриті API Redfish DMTF для компонуємих (composable) інфраструктур, що забезпечує інтероперабельність платформи Red Hat з Intel Rack Scale Design (Intel RSD) і іншими галузевими рішення з підтримкою API Redfish.

Мережева автоматизація з використанням OpenDaylight

В Red Hat OpenStack 12 подальший розвиток одержала technology preview версія OpenDaylight, модульної платформи з відкритим кодом, призначеної для настроювання й автоматизації програмно-визначаємих мереж (SDN). У прагненні допомогти замовникам збільшити швидкість і пропускну здатність мереж, Red Hat неухильно розширює підтримку віртуалізації мережевих функцій (NFV) у рамках системи OpenDaylight для OpenStack. Зокрема, компанія не тільки оптимізувала реалізацію Data Plane Developer Kit (DPDK) у рамках своєї версії OpenDaylight, але й підвищила продуктивність за рахунок можливостей SDN, відзначили в Red Hat.

Distributed Continuous Integration

Раніше Red Hat представила Distributed Continuous Integration (DCI) – механізм взаємодії замовників і партнерів з Red Hat OpenStack Platform. Основна мета, що Red Hat переслідувала при запуску DCI, полягала в тім, щоб зробити свою програмну платформу OpenStack оптимальної за рахунок автоматизації завдань розгортання, тестування й зворотного зв'язка із замовниками й партнерами на етапах до й після випуску релізів ПЗ. Таким чином, Red Hat одержала можливість тестувати і контролювати роботу своєї платформи на прикладах реальних замовників і компаній-партнерів. Інформація, що збирається DCI, допомагає інженерам Red Hat скоротити час, необхідне для ідентифікації, розробки, тестування й випуску програмних виправлень. Red Hat OpenStack Platform 12 доступна на порталі Red Hat Customer Portal, а також у складі рішень Red Hat Cloud Infrastructure і Red Hat Cloud Suite.

КОПЗ_2026

Red Hat OpenStack Platform 11

Версія IaaS-платформи OpenStack Platform 11. Вона створена на основі релізу Ocata – відкритого проекту OpenStack і пропонує механізм відновлення компонентів платформи з використанням складених ролей, розширені мережеві можливості, поліпшену інтеграцію із системою управленияRed Hat CloudForms.

В Red Hat OpenStack Platform 11 поліпшена інтеграція з Red Hat Ceph Storage, створена можливість розгорнути її на обчислювальному вузлі OpenStack (раніше ця була можливо у версії techpreview). Ceph Storage може забезпечувати для OpenStack дзеркалювання сховищ даних і, як наслідок, реплікацію між сайтами для аварійного відновлення.

Механізм відновлення компонентів платформи

У попередній версії Red Hat OpenStack Platform з'явилися складені ролі – набори з окремих сервісів, формовані оператором. За допомогою таких ролей можна прицільно масштабувати певні сервіси без масштабування хмари в цілому, що спрощує й прискорює настроювання системи у відповідь на зміну потреб підприємства. Одночасно складені ролі ускладнили відновлення компонентів платформи. Для рішення цієї проблеми в Red Hat OpenStack Platform 11 створений механізм відновлення складених ролей, що забезпечує гнучке й погоджене розгортання й відновлення компонентів.

Компонування служб OpenStack можуть виконуватися й призначатися в індивідуальному порядку, це дозволяє операторам розміщати компоненти – бази даних, проксі-сервіси або служби обміну повідомленнями на конкретних вузлах. Додавати складені ролі в раніше розгорнуту, працюючу хмару, що підвищує якість і гнучкість обслуговування замовників і партнерів у міру розширення їхніх потреб у хмарних технологіях.

Розширена підтримка мережі й NFV

Керування мережами й віртуалізація компонентів мережі (NFV) залишаються значимими функціями OpenStack, і в OpenStack Platform 11 пропонується ряд можливостей, які можуть використовуватися в стандартних корпоративних сценаріях і масштабних конфігураціях постачальників

КПЗ_2026

комунікаційних послуг або IT-сервісів. В VLAN-aware віртуальних машинах тепер повністю підтримуються прийом і відправлення пакетів з VLAN-Мітками віртуальними машинами в OpenStack при використанні Open vSwitch (OVS) або OVS Data Plane Development Kit (OVS-DPDK). Функції й можливості для телеком-провайдерів, реалізовані в оновлених версіях OVS і DPDK, підвищують продуктивність системи.

Поліпшена система керування Red Hat CloudForms

Поліпшена інтеграція із системою керування Red Hat CloudForms полегшує повсякденну експлуатацію Red Hat OpenStack Platform 11. Зокрема, Red Hat CloudForms, що входить до складу Red Hat OpenStack Platform 11, тепер підтримує контроль і керування регіонами OpenStack, декількома доменами, агрегованими вузлами (host aggregates), забезпечуючи уніфікований підхід до відображення стану хмари і його ефективності. Крім того, у версії поліпшене керування сховищами даних, зокрема, Red Hat CloudForms підтримує операції зі знімками, такі як створення, відображення й видалення.

Особливості життєвого циклу продукту

У попередній версії Red Hat OpenStack Platform представлена модель життєвого циклу продукту, відповідно до якої версії діляться на довгострокові й проміжні. Кожна третя версія Red Hat OpenStack Platform, починаючи з 10-ой, є довгостроковою й забезпечується підтримкою строком до п'яти років. Інші версії, і в їхньому числі Red Hat OpenStack Platform 11 – проміжні й забезпечуються підтримкою в один рік, включаючи підтримку з питань міграції й відновлення до наступних версій продукту. Red Hat OpenStack Platform 11 доступна для завантаження на порталі Red Hat Customer Portal у вигляді самостійного продукту або в складі технологій Red Hat Cloud Infrastructure і Red Hat Cloud Suite.

Red Hat OpenStack Platform 10

Red Hat OpenStack Platform 10 – хмарне рішення класу IaaS з високим рівнем масштабованості й адаптивності. Маючи у своїй основі СПО OpenStack версії Newton, рішення Red Hat пропонує ряд функцій, які поліпшують масштабованість системи, полегшують керування інфраструктурою, розширюють

КПІВ_2026

можливості керування, збільшують продуктивність мережевих операцій і підвищують безпеку. Життєвий цикл Red Hat OpenStack Platform 10 забезпечує підтримку продукту строком до 5 років.

Red Hat OpenStack Platform діє в основі гібридних і приватних хмарних середовищ, що забезпечують роботу критично важливих систем сотень корпоративних замовників, серед яких фінансова група BBVA, Кембриджський університет, розроблювач фінансово-аналітичних рішень FICO, Лабораторія реактивного руху НАСА, онлайн-букмекер Paddy Power Betfair, оператори Swisscom і Verizon, постачальник хмарних послуг UKCloud. Ще одна сильна сторона платформи Red Hat – потужна партнерська екосистема, у складі якої постачальники сертифікованих рішень корпоративного класу Cisco, Dell EMC, Fujitsu, Intel, Lenovo, Rackspace і NetApp, Ericsson, Nokia, NEC,Huawei і інші провідні представники телеком-індустрії.

Red Hat OpenStack Platform 10 пропонує замовникові надійну хмарну платформу, в основі якої Red Hat Enterprise Linux. Являючи собою посилений і дороблений варіант СПО OpenStack, рішення Red Hat забезпечує високий рівень адаптивності й масштабованості, дозволяючи оперативно реагувати на зміну потреб і запитів замовника, не поступаючись доступністю, продуктивністю й безпекою. До складу рішення Red Hat OpenStack Platform також входить Red Hat CloudForms, платформа керування й моніторингу для гібридних хмарних середовищ, що забезпечує контроль як над компонентами інфраструктури OpenStack, так і над робочими навантаженнями, що виконуються в хмарі OpenStack. Додатково надається 64ТБ ємності програмно-визначаємої системи зберігання даних Red Hat Ceph Storage, що дозволить оцінити потенціал гнучкої й гіпермасштабованої системи Ceph.

У складі Red Hat OpenStack Platform 10 пропонуються ключові нововведення й поліпшення:

– Полегшена робота із системою за рахунок простого й зручного графічного інтерфейсу для установки й супроводу системи. Графічний інтерфейс

КПЗ_2026

Red Hat OpenStack Platform Director одержав ряд нових функцій, таких як автоматичне відновлення модулів (включаючи Ceph), розширена настроювання мережевих конфігурацій і засобів високої доступності, а також розгортання загального доступу до файлів засобами сервісу Manila з використанням інтегрованих драйверів NetApp і Red Hat Ceph Storage.

– Більше масштабованості за рахунок додаткової гнучкості, що забезпечується із введенням сервісів, що набудовуються, і адміністративних ролей, (створюються за допомогою Red Hat OpenStack Platform Director). Версія дозволяє операторам хмарних середовищ OpenStack виконувати масштабування своїх середовищ і набудовувати в них запуск сервісів незалежно друг від друга, що підвищує гнучкість при розгортанні служб, необхідних для того або іншого робочого навантаження.

– Поліпшений захист даних за рахунок удосконалених засобів високої доступності для великомасштабних середовищ і інших доробок. Опціональне шифрування об'єктного сховища й тимчасові маркери безпеки підвищують захищеність і знижують ризики несанкціонованого доступу до даних.

– Ріст продуктивності навантажень, що інтенсивно працюють із мережею. Досягається за рахунок нового компонента DPDK (Data Plane Developer Kit) у складі модуля Open vSwitch, а також технології SR-IOV (Single-root Input/Output Virtualization), завдяки яким продуктивність по обробці мережевого вводу-виводу виходить на високий рівень. Мережева маршрутизація стала стала більше швидкої й гнучкої, з можливістю вибору централізованої або розподіленої реалізації (DVR).

– Програма сертифікації устаткування Ready-state, що базується на проекті OpenStack Ironic. Red Hat і раніше підтримувала сертифіковані апаратні плагіни для деяких компонентів OpenStack, однак програма допомагає розширити екосистему апаратних рішень для автоматизованого розгортання на серверах без передвстановленого ПЗ. Першим партнером Red Hat, що пройшов сертифікацію

КБПЗ-2026

Red Hat OpenStack Platform 10 Ready-state, стала компанія Dell EMC, найближчим часом цей список поповниться ще декількома виробниками устаткування.

Життєвий цикл із підтримкою строком до 5 років

Red Hat OpenStack Platform 10 пропонує дві моделі життєвого циклу продукту. У рамках першої моделі замовникові пропонуються довгострокові версії Red Hat OpenStack Platform із трирічним основним життєвим циклом і підтримкою, що здобувається додатково, на наступні два роки. Друга модель орієнтована на ті, хто зацікавлений у регулярному відновленні платформи до останньої версії раз у півроку або раз у рік.

Довгостроковим буде кожний третій реліз, починаючи з Red Hat OpenStack Platform 10. Інші версії будуть мати життєвий цикл підтримки протягом одного року.

Red Hat OpenStack Platform 9

Версія 9 платформи Red Hat створена на основі релізу OpenStack Mitaka і пропонує готову до використання автоматизовану хмарну платформу з посиленим захистом, що інтегрується з Red Hat Enterprise Linux 7.2,Red Hat Ceph Storage 2 і Red Hat CloudForms, забезпечуючи комплексне керування й моніторинг у гібридних хмарних середовищах.

У складі Red Hat OpenStack Platform 9 пропонуються технологічні вдосконалення на рівні майже всіх служб OpenStack:

– Автоматичне відновлення ПЗ з використанням Red Hat OpenStack Platform Director – рішення Red Hat дозволяє виконувати відновлення середовища OpenStack з використанням механізмів автоматизації й перевірки інструментарію Red Hat OpenStack Platform Director, що побудований на основі відкритого проекту Triple (OpenStack on OpenStack) і значно спрощує перехід на нові версії OpenStack при мінімізації простою середовищ, що працюють у режимі промислової експлуатації.

– Живаючи міграція й тверда прив'язка до ЦП засобами OpenStack Compute (Nova) – компонент Compute забезпечує прискорений і поліпшений

КОПЗ–2026

перенос працюючого екземпляра ВМ, дозволяючи адміністраторові візуально контролювати хід процесу й припиняти й відновляти завдання по виконанню міграції. Функція твердої прив'язки віртуальних ЦП до фізичних процесорних ядер може динамічно міняти поводження гіпервізору для вимогливих до затримок навантажень, таких як NFV, забезпечуючи більше тонке керування продуктивністю.

– Попередня версія драйвера резервного копіювання Google Cloud Storage для OpenStack Block Storage (Cinder) – Red Hat OpenStack Platform 9 продовжує співробітництво з компанією Google і пропонує політики аварійного відновлення, що підтримують загальнодоступне хмарне сховище Google Cloud Storage за допомогою убудованого драйвера, що допомагає організувати додатковий рівень захисту критично важливих даних у гібридній хмарі.

Red Hat OpenStack Platform 9 допомагає скоротити строки запуску сервісів і створити функціонал самообслуговування для системних адміністраторів при організації приватних хмарних середовищ OpenStack за допомогою підключення системи керування Red Hat CloudForms і організації єдиного оточення з високим ступенем автоматизації. Red Hat CloudForms входить до складу підписки Red Hat OpenStack Platform і пропонує засобу виявлення, моніторингу й глибокої перевірки ресурсів OpenStack, які дозволяють організувати керування життєвим циклом і експлуатацію всіх компонентів інфраструктури OpenStack, а також працюючих у цьому середовищі віртуальних робочих навантажень, на основі політик. Модернізований Red Hat OpenStack Platform Director забезпечує розгортання програмного сховища Red Hat Ceph Storage, що використовується в приватних хмарних середовищах Red Hat OpenStack Platform і пропонує клієнтові масштабоване хмарне сховище з безкоштовною ємністю 64 ТБ.

Red Hat OpenStack Platform 8

Red Hat OpenStack Platform 8 – версія платформи OpenStack від Red Hat, до складу якої входить програмна СХД Red Hat Ceph Storage і інструментарій керування Red Hat CloudForms.

КПЗ_2026

Приватні хмарні рішення на основі OpenStack від Red Hat – це відкрита платформа промислового рівня, що поєднує високо масштабовану інфраструктуру із засобами керування й технології, що підвищують продуктивність розроблювачів, а також має велика екосистему сертифікованих програмних і апаратних партнерських рішень. Доступні як платформа Red Hat OpenStack Platform або у вигляді інтегрованого пакета Red Hat Cloud Suite, хмарні рішення Red Hat забезпечують побудова масштабованих відказостійких середовищ IaaS. В основі гібридних хмарних рішень Red Hat лежить Red Hat OpenStack Platform 8, версія високо масштабованої платформи IaaS, що являє собою дороблений компанією Red Hat варіант вільно розповсюджуваного ПЗ OpenStack версії Liberty, об'єднаний з операційною системою Red Hat Enterprise Linux для одержання готової до промислового використання й яка відповідає найсучасніших вимог хмарної платформи. Платформа Red Hat одержала визнання в телекомунікаційній і IT-галузі, що ведуть представники якої об'єднаються навколо її для переходу до програмно-визначаємих мереж (SDN) і віртуалізації мережевих функцій (NFV).

З випуском Red Hat OpenStack Platform 8 компанія Red Hat пропонує платформу OpenStack, що штатно підтримує наступні функції:

– Автоматичне відновлення – компонентів Red Hat OpenStack Platform Director забезпечує простий перехід на нові основні й проміжні версії, автоматично виконуючи необхідні відновлення в масштабах системи як для основних сервісів OpenStack, так і для самого компонента Director. Допомагає підтримувати хмара OpenStack у справному й стабільному стані при мінімальних простоях.

– Керування інфраструктурою й робочими навантаженнями – інструментарій Red Hat CloudForms забезпечує ефективне керування інфраструктурою OpenStack і робітниками навантаженнями Linux і Windows, включаючи керування життєвим циклом навантажень, моніторинг, звіти, мульти-вузлову оркестрацію, регулювання й керування на основі політик.

КОПЗ – 2026

– Програмно-визначаєме сховище – високо-масштабируема програмна СЗД Red Hat Ceph Storage, що тепер входить до складу Red Hat OpenStack Platform, забезпечує підтримку гнучкого об'єктного й хмарного сховища ємністю 64 ТБ. Є самої популярної СЗД для хмарних середовищ OpenStack , надаючи універсальну платформу для підтримки зростаючих обсягів даних у таких середовищах.

– Спеціальні функції для телеком-операторів (попередні версії) – Red Hat OpenStack Platform 8 містить ряд нових і важливих функцій, що стосуються віртуалізації мережевих компонентів, включаючи режим реального часу для гіпервізору KVM (підвищує передбачуваність затримок), підтримку бібліотек і драйверів DPDK на рівні комутатора Open vSwitch (підвищує продуктивність мережевого вводу-виводу), плагін OpenDaylight, що забезпечує можливість побудови програмно-визначаємих мереж.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об’єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ. Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python https://www.python.org/ і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію. Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для налаштовуваних програм.

КБПЗ – 2026

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КОПЗ_2026

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Відмінності від гіперконвергентної архітектури

Як і в конвергентних і гіперконвергентних інфраструктурах, у компонентній поєднуються обчислювальні потужності, ємності зберігання й мережеві ресурси, однак попереднє конфігурування для підтримки певних класів навантажень відсутній.

Гіперконвергентні системи більше підходять для масштабування однотипних навантажень, коли на всіх вузлах виконуються однакові застосунки. Вузли мають фіксовану конфігурацію, що спрощує масштабування HCI, але обмежує її гнучкість.

В HCI обчислювальні потужності й ресурси зберігання масштабуються блоками, тоді як у компонуємій інфраструктурі вони дезагреговані. Це дозволяє забезпечити більш гнучке масштабування за рахунок незалежного нарощування обчислювальних ресурсів і ємності зберігання.

Якщо гіперконвергенція забезпечує горизонтальне масштабування за рахунок агрегування ресурсів у фіксовані блоки, то дезагрегована архітектура (Rack Scale Design, RSD) використовує для цієї мети, відповідно, дезагрегацію. Компонуєма ж архітектура намагається усунути розходження між цими двома, визначаючи агрегацію на програмному рівні, а не на апаратному.

Компонуєма інфраструктура дозволяє надавати додаткам комп'ютерні ресурси за допомогою коду й, таким чином, рятує від необхідності фізично міняти конфігурацію устаткування при додаванні або відновленні застосунків. Тому вона нерідко розглядається як реалізація концепції «інфраструктура як код».

КБПЗ_2026

«Інфраструктура як код» припускає використання готових сценаріїв для автоматизації розгортання інфраструктури на вимогу. Для цього використовуються, зокрема, продукти Ansible і Chef. Постачальники компонуємих систем використовують ту ж саму концепцію для спрощення розгортання інфраструктури на базі наявного пула ресурсів. Відповідно, при описі своїх систем вони нерідко використовують визначення «дезагрегований» і «інфраструктура як код» (Infrastructure as a Code, IAC). Існуючі компонуємі системи являють собою, по суті, об'єднання дезагрегованой апаратної платформи й програмної надбудови IAC.

CIS у контексті програмувальних ЦОД

У звіті Forrester Research, озаглавленому «Програмно-визначаємі ЦОД досягли зрілості» («The Software-Defined Data Center Comes of Age», Forrester Research, 2026), затверджується, що в майбутньому компонуємі інфраструктурні системи (Composable Infrastructure Systems, CIS) стануть переважним підходом до реалізації обчислювальної інфраструктури в програмувальних центрах (Software-Defined Data Center, SDDC).

Компонуєма інфраструктура являє собою одну з можливих реалізацій програмно-визначаємой середовища (Software Defined Infrastructure, SDI). Поява програмно-визначаємих серверів логічно доповнює програмно-визначаємі зберігання й мережі, тому що вони можуть служити як рівень, на базі якого дані сервіси можуть надаватися у вигляді «компонуємих корисних навантажень» усередині віртуальних машин або контейнерів.

Для внесення змін у конфігурацію устаткування ЦОД необхідна участь людини: подача заявки, її схвалення, виконання, – на все це потрібен час. В ідеалі, коли повністю компонуємі центри обробки даних стануть реальністю, адміністраторам не прийде визначати вимоги до ресурсів для підтримки тих або інших застосунків – це буде робитися автоматично.

Керуюче програмне забезпечення ЦОД зможе виділяти необхідні ресурси безвідносно до тому, де фізично вони перебувають. Рятування від необхідності

КОПЗ – 2026

людського втручання дозволить не тільки прискорити впровадження змін, але й уникнути помилок і підвищити безпека

У цьому ідеальному ЦОД низькорівневі механізми контролю будуть дезагрегувати і потім агрегувати ядра, пам'ять, зберігання й мережу таким чином, щоб застосунки одержували необхідну конфігурацію поза залежністю від реального компонування фізичної інфраструктури.

Однак інфраструктурні блоки, з яких формуються ЦОД, поки недостатньо гнучкі: процесор і пам'ять тісно прив'язані до системної плати сервера, тому в такій ситуації виконати їхню агрегацію не так просто, ніж у випадку, коли вони перебували б на різних серверах (наприклад, на зразок жорстких дисків).

Як відзначалося вище, щоб повністю компонуєма інфраструктура й, відповідно, повністю компонуємі центри даних стали реальністю, необхідно, щоб центральним елементом обчислювальної архітектури стала пам'ять, а не процесор (Memory-Centric Architecture, MCA). У той же час архітектуру, орієнтовану на роботу з пам'яттю, в IDC розглядають як наступну фазу розвитку архітектури після компонуємої/дезагрегованої архітектури.

Хто є хто на ринку CIS

Термін «компонуєма інфраструктура» уведений у широкий оберт компанією HPE, що представила відповідне рішення в 2015 році. В 2025 році, як випливає зі звіту Forrester, HPE залишалася єдиним вендором, чиє рішення – HPE Synergy – відповідало визначенню локальної компонуємої інфраструктурної системи.

За твердженням цього виробника, HPE Synergy можна повністю запрограмувати з використанням єдиного API. Пропонований «інноваційний компонуємий інтерфейс прикладного програмування» дозволяє «за допомогою одного рядка коду» описати й ініціювати необхідну застосунку інфраструктуру без трудомісткого написання сценаріїв.

Тим часом в Cisco вважають, що рішення HPE не відповідає вимогам до компонуємої інфраструктури. У технічному бюлетені компанії, озаглавленому

КПЗ_2026

«HPE Synergy: претензії на компонуємость недостатньо, щоб сховати складну інфраструктуру», затверджується, що в цьому випадку необхідно знати заздалегідь, які ресурси знадобляться застосунку, а після розгортання системи її важко змінити, масштабувати або перепрофілювати.

Відповідні рішення Cisco з'явилися приблизно в той же час, що й в HPE, але незабаром компанія відмовилася від просування серверів Cisco UCS M-Series, які вона позиціонувала як компонуємі системи. Як були заявлено, «інвестиції перенаправляються на програми, які дозволять реалізувати ці можливості у всіх рішеннях UCS».

У свою чергу, автори технічного бюлетеня «HPE Synergy у якості компонуємої інфраструктури в порівнянні з Cisco UCS» затверджують, що HPE Synergy являє собою більше уніфіковане, чим UCS, рішення в змісті об'єднання обчислювальних потужностей, ємностей зберігання й мережевих ресурсів, при цьому воно простіше й дешевше в експлуатації й реалізує найсучаснішу – компонуєму – архітектуру.

Тим часом на ринку з'явився третій претендент. Як прогнозували у своєму звіті автори Forrester, концепція компонуємої інфраструктури настільки приваблива, що в 2025 році можна чекати хвилю оголошень про випуск таких систем. Дійсно, навесні Dell EMC представила Kinetic Infrastructure.

Для опису своєї компонуємої архітектури в Dell EMC використовують визначення «кінетичний». Цим хочуть підкреслити той факт, що пропоноване рішення не обмежується модульним дизайном, а принцип компонуємості поширюється на окремі пристрої зберігання – а в остаточному підсумку на пам'ять і інші залежні пристрої, зокрема DRAM, SCM, GPU і FPGA.

Крім визнаних вендорів, свої рішення пропонують і декілька стартапів. Однак число їх невелике, що підкреслює той факт, що цей сегмент ринку тільки починає формуватися. До того ж, як і Dell EMC, вони нерідко використовують іншу термінологію для опису своїх рішень і інші принципи їхньої реалізації.

КПЗ – 2026

Так, TidalScale називає своє рішення «програмно-визначаємим сервером». За твердженням розроблювача, його технологія дозволяє об'єднати кілька серверів в одного віртуальний практично будь-якого розміру. Процесори, пам'ять і вводу-вивід поєднуваних систем агрегуються, при цьому немає необхідності вносити які-небудь зміни у використовувані ОС і застосунки.

Говорячи про компонуєму архітектуру, не можна не згадати Intel. В 2013 році компанія запропонувала концепцію масштабованої архітектури (Rack Scale Architecture, RSA), що припускає, що типові компоненти серверів поєднуються в пули, а не розташовуються на системній платі. У результаті компонента виявляються більше не прив'язані до конкретного сервера і їх можна обновляти незалежно.

Intel Rack Scale Design (так тепер називається Rack Scale Architecture) реалізує концепцію компонуємої дезагрегованої архітектури (Composable Disaggregated Infrastructure, CDI). Крім іншого, він визначає стандартний набір API, які Intel запропонувала як розширення платформи керування RESTful Redfish. Відкритість пропонованої архітектури дозволяє уникнути залежності від вендору й гарантує стабільність і інтероперабельність.

Рік назад OpenStack Foundation випустила 16-ю по рахунку версію хмарної інфраструктурної платформи на базі відкритого коду за назвою Pike, у якій акцент зроблений на компонуємих сервісах. Як затверджується в анонсі, компонуємість робить можливими такі сценарії використання, як граничні обчислення й NFV, при цьому модульна архітектура дозволяє вибирати, яку функціональність включати в інфраструктурний стек, будь те блокове сховище або «чисті» сервери.

Висновок

Обчислювальна інфраструктура існуючих центрів обробки даних не має той ступінь гнучкості, яка необхідна для підтримки всього розмаїтості можливих навантажень, при цьому компонуєма інфраструктура забезпечує таку підтримку одночасно й для успадкованих, і для перспективних навантажень – «чистих»

КПЗ-2026

серверів, віртуальних машин і контейнерів. Компонуємі системи дозволяють оптимізувати використання існуючої IT-інфраструктури за рахунок програмно-визначаємой агрегації ресурсів, віртуалізації наявних інфраструктурних активів і програмного керування ними. У результаті підвищується ступінь використання ресурсів, знижуються капітальні й операційні витрати, з'являється можливість швидко адаптувати інфраструктуру до мінливих навантажень.

Як вважають в IDC, якщо не відбудеться ніяких проривів у кремнієвій фотоніці, те програмно-визначаєма компонуєма архітектура буде розвиватися швидше, ніж апаратна дезагрегація. Удосконалювання останньої створить умови для її включення в компонуєму архітектуру, можливості якої після цього ще більше розширяться.

3.2 Розробка структурної схеми

Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі – програмно-визначаєме рішення, що дозволяє управляти інфраструктурою як кодом, здійснювати швидке розгортання ресурсів для будь-якого робочого завдання.

Основою системи виступають програмно-визначаємі технології, а для рішення конкретних завдань використовуються готові шаблони. Інформаційні ресурси об'єднані в гнучкі кластери, керування здійснюється за допомогою універсальних інтерфейсів. Можливості системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі оптимальні для застосування в області DevOps (командна робота розроблювачів і IT-підрозділу) і гібридної хмари.

Розроблене програмне забезпечення системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі взаємодіє з продуктами Hewlett Packard Enterprise (HPE). Нижче розглянемо з якими саме програмними продуктами HPE відбувається взаємодія.

КБПЗ 2026

Інтеграція системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі з HPE Primera

Hewlett Packard Enterprise (HPE) оголосила про інтеграцію ключових елементів свого портфеля, які сполучать у собі можливості штучного інтелекту (ШІ) і компонуємої інфраструктури. Компанія інтегрувала платформу для зберігання даних HPE Primera, що опирається на ШІ, з компонуємою платформою Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, щоб допомогти замовникам швидше розгортати нові застосунки й розробляти інновації для розвитку свого бізнесу. Докладніше тут.

Інтеграція системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі з VMware Cloud Foundation

Hewlett Packard Enterprise (HPE) оголосила про розширення співробітництва з компанією VMware, у рамках якого замовники одержать доступ до VMware Cloud Foundation по моделі «як послуга». Завдяки інтеграції HPE GreenLake і Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі з VMware Cloud Foundation, компанії дадуть своїм загальним замовникам можливість зберігати всі свої застосунки й дані локально, одночасно користуючись перевагами хмарної й компонуємої інфраструктури. Докладніше тут.

Відкриття доступу системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі до функцій HPE InfoSight

Hewlett Packard Enterprise (HPE) оголосила про відкриття доступу до функціями HPE InfoSight – інструмента керування інфраструктурою за допомогою штучного інтелекту (ШІ) – для серверів HPE ProLiant, обчислювальних модулів Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі і систем HPE Apollo. Користуючись можливостями глобальної платформо-передбачувальної

КБПЗ-2026

аналітики, що самонавчається на основі реальних даних про функціонування сотень тисяч установлених по усьому світі продуктів HPE, HPE InfoSight, зі слів розроблювача, знижує операційні витрати, підвищує продуктивність IT-систем і вивільняє час, що IT-фахівці можуть використовувати для створення інновацій.

Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі – платформа для побудови компонуємої інфраструктури. В одному корпусі об'єднані модулі зберігання, мережеві, обчислювальні компоненти й програмне забезпечення. Система дозволяє за кілька хвилин розгорнути будь-який додаток: від ERP-систем, серверів електронної пошти, баз даних до зовсім нових, написаних для хмарного середовища.

З переваг платформи відзначимо зручне підключення до мереж ЦОД, єдину мережеву фабрику, що включає до п'яти шасі Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, що дозволяє заощадити на мережевих підключеннях, і схему комутації Master/Satellite. Зв'язок трьох фреймів забезпечують конвергентні модулі Virtual Connect, підключені за такою схемою, що скорочує число зовнішніх комунікаційних з'єднань.

У цілому при роботі з розробленим у даній роботі програмним забезпеченням можна познайомитися з архітектурою Система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі і протестувати:

– інтегровану систему керування Composable Infrastructure System Composer на базі OneView 3 для централізованого комплексного керування всіма компонентами;

– систему керування образами й масовим розгортанням Composable Infrastructure System Image Streamer;

– роботу Composable Infrastructure System у конфігурації із системою зберігання даних 3Par;

КОПЗ – 2026

– роботу обчислювальних модулів з інтегрованою системою зберігання прямого підключення D3940.

Все описане вище знайшло своє відображення у структурній схемі.



Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Однією з основних функцій системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі є забезпечення QoS. Quality of Service (QoS) – технологія надання різним класам трафіку різних пріоритетів в обслуговуванні – якості обслуговування. Іншими словами це технологія, що забезпечує виділення переваг високопріорітетному мережному трафіку, пристрою або критичному застосунку. Під якістю обслуговування (QoS) розуміється здатність мережної інфраструктури надавати поліпшене обслуговування певному виду переданого трафіку (пакетів) за допомогою різних технологій.

КПЗ_2026

Існує дві основні архітектури QoS:

– Інтегроване обслуговування (IntServ).

– Диференційовані послуги (DiffServ).

Integrated Service – інтегроване обслуговування забезпечує наскрізне (End-to-End) якість обслуговування, тобто відбувається резервування ресурсів на всьому шляху проходження трафіку. Для резервування ресурсів (Resource reservation) використовується протокол RSVP, гарантуючи необхідну пропускну здатність. Протокол резервування ресурсів (RSVP) компанія Microsoft запропонувала для QoS в Windows 2000. Істотним недоліком є постійне резервування ресурсу, навіть у тому випадку, якщо він не використовується або використовується не повністю.

Диференціальний вид послуг припускає наявність певного набору засобів класифікації й механізмів організації черг, що забезпечують роботу із пріоритетами й виконує дві функції:

– Формування трафіку на границях мережі – функції класифікації, маркування пакетів і керування інтенсивністю.

– Політика покрокового обслуговування PHB (Per-Hop Behavior) включає функції розподілу ресурсів і відкидання пакетів.

Якість обслуговування на другому рівні моделі OSI (канальному) у межах одного мережного елемента забезпечується за рахунок використання моделі диференційованого обслуговування (Differentiated Service – DiffServ) і забезпечується:

– Класифікацією й розміткою трафіку.

– Керуванням перевантаженнями (механізми черг).

Слід зазначити, що дана модель починає працювати лише у випадку появи черг і перевантажень.

Класифікація пакетів (Packet Classification) – віднесення пакета до певного класу.

Маркування пакетів (Packet Marking) – установка необхідного пріоритету.

КОПЗ-2026

Слід зазначити, що класифікація й маркування пакетів відрізняються залежно від рівня OSI, на якому працює пристрій. Як правило, всі комутатори працюють на рівні L2, а саме з Ethernet кадрами (фреймами). Маршутизатори й деякі просунуті комутатори працюють на рівні L3 і вже не з кадрами, а пакетами.

Протокол Ethernet у чистому виді не підтримує функцію пріоритезації трафіку, тому поряд зі стандартним протоколом Ethernet IEEE 802.3, організація IEEE розробила стандарт створення віртуальних локальних мереж VLAN IEEE 802.1q. У стандарті IEEE 802.1q передбачається вставка додаткового чотирьохбайтового тегу VLAN у заголовок Ethernet вихідного фрейму, що містить мітку пріоритету (Priority) класу обслуговування (Class of Service – CoS) IEEE 802.1p.

Нижче перераховані типи трафіку, починаючи з високо пріоритетного:

– Керування мережею (7). Передача даних для підтримки мережної інфраструктури (кадри маршрутних протоколів).

– Голос (6). Критичний по затримці (< 10 мсек) при інтерактивних переговорах.

– Відео (5). Критичний по затримці (< 100 мсек) при інтерактивних відео обмінах.

– Контрольоване навантаження (4). Робота в ситуації некритичної по затримці, але критичної по втратах (наприклад, діловий трафік, потоковий трафік з резервуванням).

– Максимальні зусилля (3). Робота в ситуації некритичної по затримці, але критичної по втратах, але в умовах з меншим пріоритетом, ніж контрольоване навантаження. У випадку інформаційної служби цей режим може використовуватися для привілейованих клієнтів.

– Найкращі зусилля (2). Це трафік звичайний трафік LAN.

– Фоновий режим (0, Background). Масові пересилання даних і будь-яка інша активність, що не впливає негативно на роботу інших.

КБПЗ – 2026

Перевантаження виникає у випадку переповнення вихідних буферів передавального трафік устаткування. Основними механізмами виникнення перевантажень (або, що рівнозначно, скупчень – congestions) є агрегація трафіку (коли швидкість вхідного трафіку перевищує швидкість вихідні) і непогодженість швидкостей на інтерфейсах. Керування пропускною здатністю у випадку перевантажень (вузьких місць) здійснюється за допомогою механізму черг. Кадри Ethernet містяться в черзі, які впорядковано обробляються по певному алгоритмі.

Фактично, керування перевантаженнями – це визначення порядку, у якому фрейми виходять із інтерфейсу (черг) на основі пріоритетів. Якщо перевантажень немає – черги не працюють.

На L3 ми маємо справу із протоколом IP (Internet Protocol). При розробці протоколу IP для цілей QoS було спеціально передбачене поле To (Type of Service) розміром один байт. До середини 90-х років поле TOS у більшості реалізацій ігнорувалося. Але після початку розробок засобів забезпечення якості обслуговування (QoS) увага до цього виросла. З'явилася пропозиція заміни поля TOS на поле DSCP (Differenciated Services Code Point), що також має 8 біт (RFC-2474). Біти CU поки не визначені. Іноді це поле називається байтом DS (Differenttiated Services).

Поле ToS може бути заповнений класифікатором IP Precedence або DSCP залежно від завдання.

IP precedence (IPP) має розмірність 3 біти, може приймати значення 0-7, тобто можна говорити про 8-мі класах обслуговування. Споконвічно використовувався класифікатор IPP, але згодом з'явилася необхідність розділяти трафік на більше чому 8 класів обслуговування, наслідком чого з'явилася розробка класифікатора DSCP.

Формат поля DSCP

Біти DS 0-DS5 визначають селектор класу. Значення цього коду представлені в таблиці нижче. Стандартним значенням DSCP за замовчуванням є 000000.

КбПЗ 2026

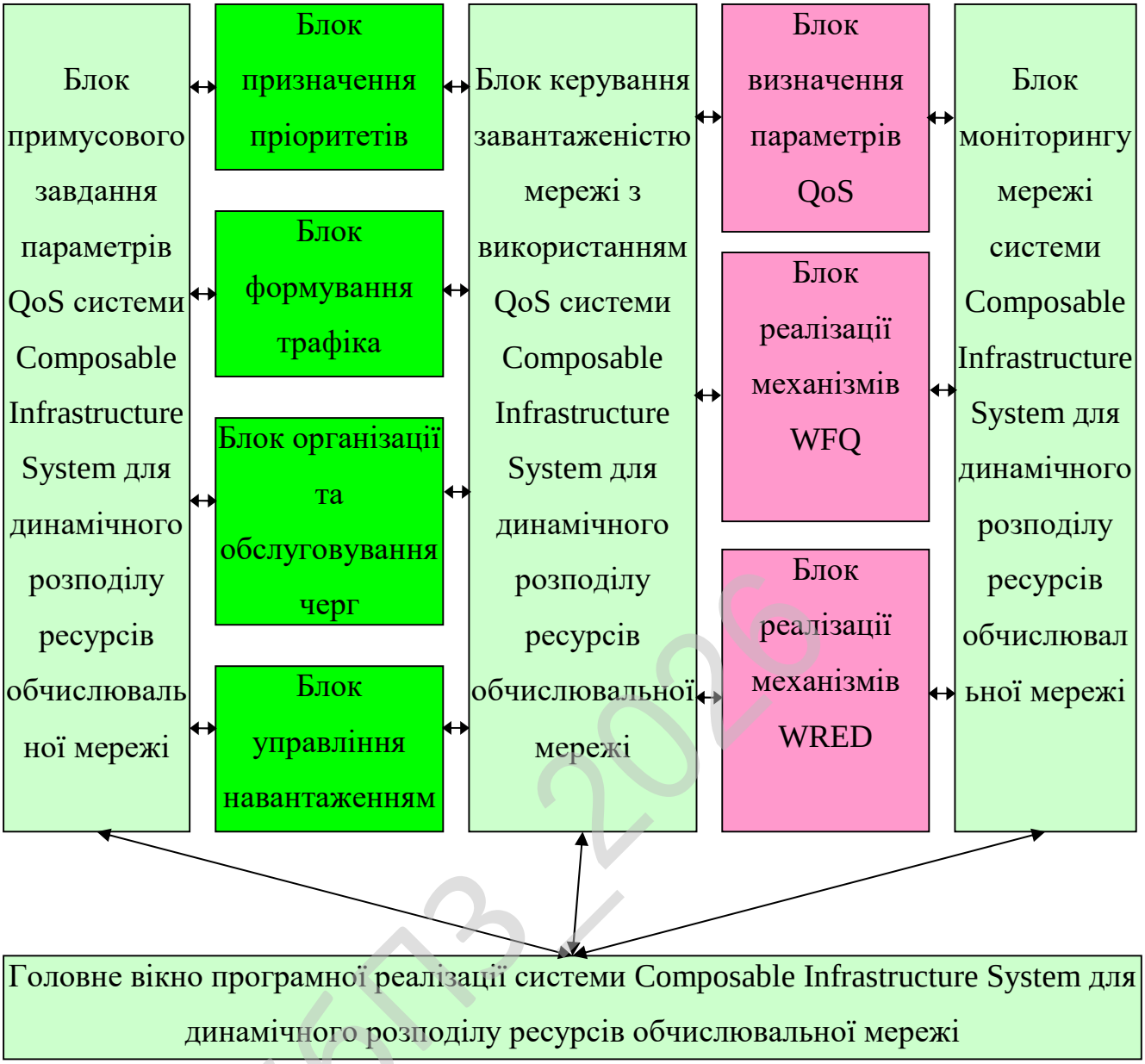
Таблиця 3.1 – Формат поля DSCP

Селектор класу	DSCP
Пріоритет 1	001000
Пріоритет 2	010000
Пріоритет 3	011000
Пріоритет 4	100000
Пріоритет 5	101000
Пріоритет 6	110000
Пріоритет 7	111000

Функціональна схема розробленої системи зображена на рисунку 3.2. Існує не занадто багато способів забезпечення QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі. Найпростіший з них – збільшення смуги пропущення мережі за рахунок нарощування апаратних можливостей устаткування. Можна використовувати й такі прийоми, як завдання пріоритетів даних, організація черг, запобігання перевантажень і формування трафіку. 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.

Збільшення апаратної потужності, безсумнівно, є найбільш ефективним засобом реалізації QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі у локальній мережі. Тиск із боку конкурентів, необхідність підвищення ефективності виробництва, поява нових технологій, що дозволяють оснащувати спеціалізовані мікросхеми (ASIC) найрізноманітнішими функціями, – все це змушує постачальників комутаційного устаткування для локальних мереж викидати на ринок усе більше швидкодіючі пристрої за цінами, порівнянним з вартістю моделей колишнього покоління.

КБПЗ_2026



КПЗ 2026

Рисунок 3.2 – Функціональна схема системи

Малоймовірно, що в доступному для огляду майбутньому даний підхід до підтримки QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі у локальних мережах перестане бути пріоритетним. Оскільки в локальних мережах вдається забезпечити гарантовану якість послуг, не прибігаючи до дорогої модернізації усього устаткування й серйозних змін у системі керування мережею, мережні адміністратори будуть звертати увагу й на програмні засоби, що дозволяють реалізувати QoS системи

Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

Отже, найбільше поширення, швидше за все, одержить комбінований підхід. Деякі виробники висловлюються на його користь, затверджуючи, що найкраще збільшувати пропускну здатність мережі не прямо, а за рахунок інтелектуальних можливостей устаткування, що має засоби забезпечення QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі. 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 пропущення 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 пропущення в глобальних мережах буде залишатися досить високою, тому й нарощування апаратної потужності не стане настільки популярним, як у локальних мережах.

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

– Головне вікно програмної реалізації системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі

– Блок моніторингу мережі системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– Блок керування завантаженістю мережі з використанням QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– Блок дослідження можливостей механізмів WRED.

– Блок дослідження можливостей механізмів WFQ.

– Блок призначення пріоритетів.

КОПЗ_2026

– Блок організації та обслуговування черг.

– Блок управління навантаженням.

– Блок формування трафіку.

– Блок інтерфейсу користувача.

– Блок визначення параметрів QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– Блок примусового завдання параметрів QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

Розглянемо ці блоки більш детально.

Головне вікно програмної реалізації системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі

Призначений для реалізації взаємодії користувача, або дослідника з системою.

Блок моніторингу мережі

Призначений для аналізу поточного стану мережі.

Блок керування завантаженістю мережі з використанням QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі

Призначений для керування завантаженістю мережі з використанням QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережіз визначеним трафіком та заданими параметрами якості обслуговування (QoS).

Блок реалізації механізмів WRED

Одним з методів QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, 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 random early detection).

КПЗ – 2026

Блок реалізації механізмів WFQ

Другим з методів QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, 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 управління навантаженням

Служба QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі 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 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 стало застосування механізму випадкового виділення пакетів (Random 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.

КОПІЯ 2026

Блок формування трафіку

Формування трафіку – це загальний термін, яким прийнято позначати різні способи маніпулювання даними для підвищення якості їхньої передачі. Один їхній таких способів – сегментація пакетів. У мережах ATM гарантовано високий рівень QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі досягається в тому числі й за рахунок малого розміру переданих пакетів (осередків – у термінології ATM). Максимальний час затримки при передачі будь-якого пакета мережі ATM – це час передачі одного осередку.

Запозичаючи корисні механізми технології ATM, 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 – його "вирівнювання". Для таких протоколів, як наприклад, AppleTalk, характерна нерівномірна передача пакетів, що часом приводить до появи в мережі послідовностей або ланцюжків пакетів, а отже – до її перевантаження. Процедура вирівнювання трафіку 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.

Блок визначення параметрів QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі

Призначений для визначення існуючих параметрів якості обслуговування (QoS). До них відносяться:

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

– Delay – затримка при передачі пакета.

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

КПЗ – 2026

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

Блок примусового завдання параметрів QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі

Призначений для примусового завдання одного, або декількох параметрів якості обслуговування (QoS). До них відносяться:

– Bandwidth (BW) – смуга пропущення, описує номінальну пропускну здатність середовища передачі інформації, визначає ширину каналу. Виміряється 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 призначення пріоритетів

Нарівні з нарощуванням апаратного забезпечення мережі для реалізації QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі застосовуються й засобу типу завдання пріоритетів даних і організації черг. Маршрутизатори підтримують ці механізми протягом багатьох років, як і деякі з нових комутаторів для каналів 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 критеріїв (наприклад, типу додатка для застосовуваного

КБПЗ_2026

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 локальної мережі, 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 дозволять 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 QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, що визначають час затримки, швидкість пропущення й рівень надійності передачі пакета. Однак найдеться небагато популярних додатків – за винятком мультимедійного ПЗ, – у які реалізована підтримка протоколу IP TOS.

Зараз розробляється протокол резервування ресурсів RSVP, що передбачає більше складний, чим в IP TOS, механізм передачі від додатка до машрутизатору 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.

КПЗ_2026

Видимо, у доступному для огляду майбутньому неявна пріоритезація, не потребуюча 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 набудовувати своє програмне забезпечення на запит найвищого з можливих рівнів послуг. Відповідно, адміністраторові мережі 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 даним призначені відповідні пріоритети (за допомогою явних або неявних методів), потрібно визначити порядок передачі цих даних, задавши алгоритм обслуговування черг із необхідною якістю (рівнем QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі). 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 з більш високим пріоритетом за умови, що й пакету з низьким пріоритетом гарантується відповідна увага. 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 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 QoS системи Composable Infrastructure System для динамічного розподілу

КБПЗ-2026

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

У більше 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 пропущення, а сама смуга пропущення 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 QoS системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі, що використовують чотири черги, хоча чим більше черг, тим більше різних пріоритетів можна привласнити переданим пакетам і тим "справедливіше" розподілити смугу пропущення між додатками. Наприклад, адміністратор у стані задати пріоритети таким чином, щоб перевага при передачі віддавалося пакетам, адресованим на більше віддалені вузли.

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

КПЗ 2026

3.4 Розробка діаграми процесів

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

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

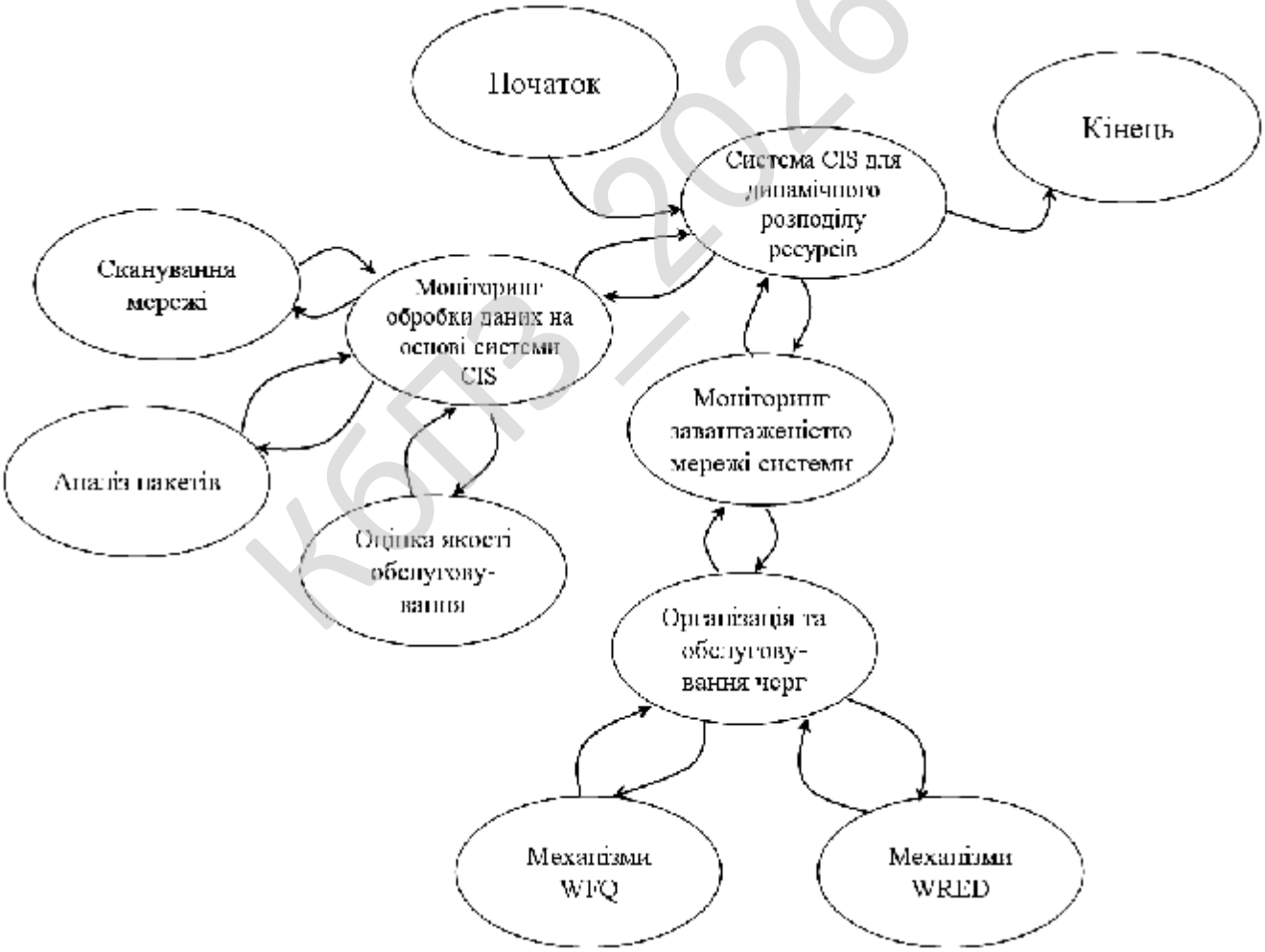


Рисунок 3.3 – Діаграма взаємодії процесів

КПІ 2026

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

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

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

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

Діаграми потоків даних містять чотири типи елементів:

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

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

КбПЗ-2026

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

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

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

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів. Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

КПЗ – 2026

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елементу.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елементу. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

КПЗ – 2026

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

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

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

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

Систе́ма керування ба́зами да́них (СКБД, Database Management System, DBMS) – набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних. Першим поколінням СКБД прийнято вважати ієрархічні й мережеві системи. Ці системи отримали широке поширення в 1970-х роках, а першою комерційною системою цього типу була система IMS компанії IBM. У 1980-х роках ці системи були витіснені системами другого покоління – повсюдно використовуваними і донині реляційними СКБД. У цих системах використовувалися непроцедурні мови управління даними (SQL) і передбачався значний ступінь незалежності даних. Реляційні системи внесли значні удосконалення в управління даними: графічний користувацький інтерфейс (GUI), клієнт-серверні застосунки, розподілені бази даних, паралельний пошук даних та інтелектуальний аналіз даних.

КПЗ 2026

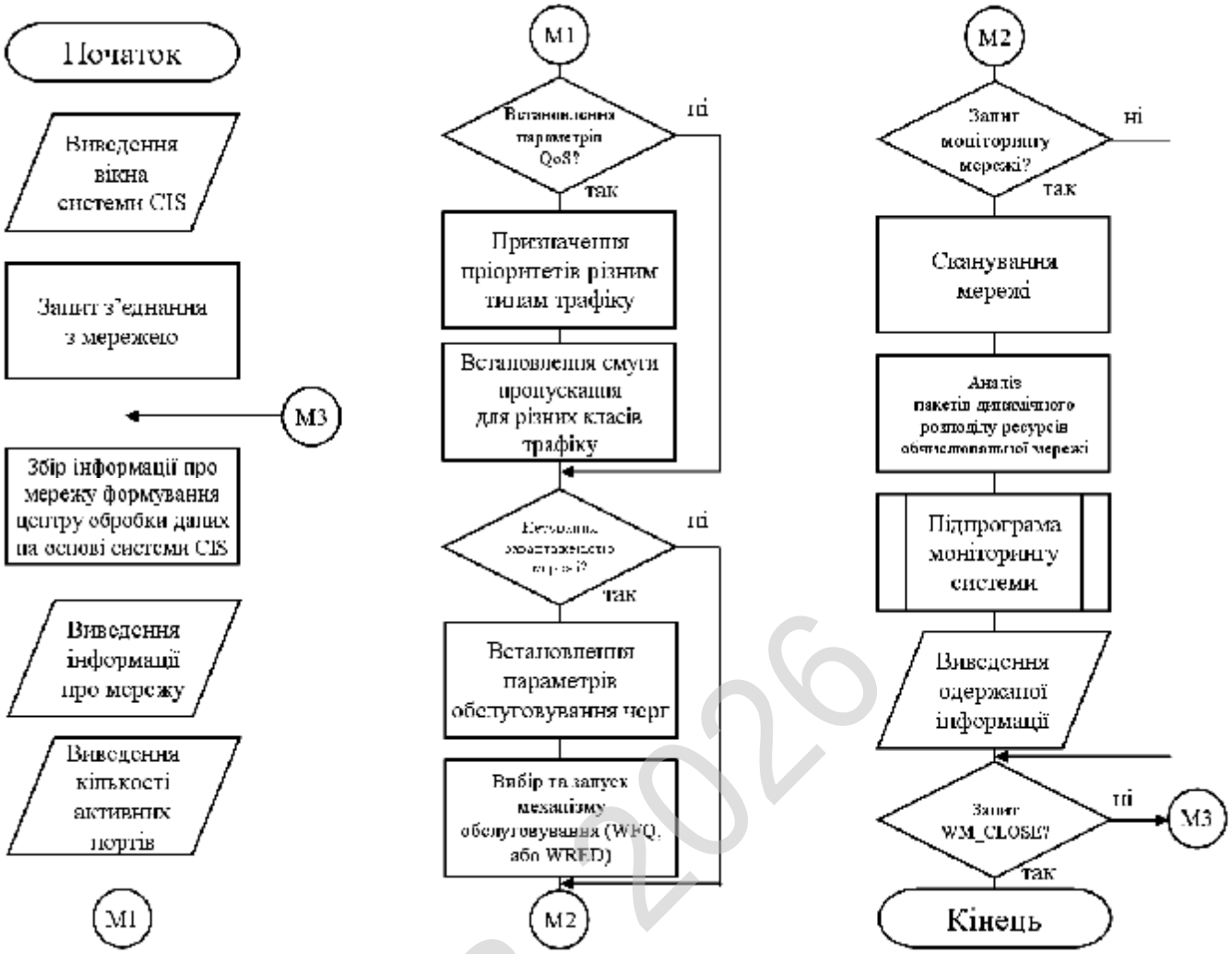


Рисунок 4.1 – Блок-схема основної програми

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

У 1991 був утворений консорціум ODMG, основною метою якого стало вироблення промислового стандарту об'єктно-орієнтованих баз даних. Між 1993 та 2001 роками ODMG опублікувала п'ять ревізій своїх специфікацій. Остання версія стандарту має індекс 3.0, після чого група розпустилася. До кінця 1990-х існувало близько десяти компаній, що виробляли комерційні продукти, що позиціонуються на ринку як ООСКБД. Найбільш відомими системами даного

КПЗ_2026

класу стали Objectivity, Versant виробництва однойменних компаній, а також СКБД Jasmine, випущена компанією CA. Незважаючи на переваги, що дозволяють ефективніше вирішувати певний ряд завдань, об'єктно-орієнтовані системи так і не змогли завоювати значущу частку ринку СКБД, залишившись «нішевим» продуктом.

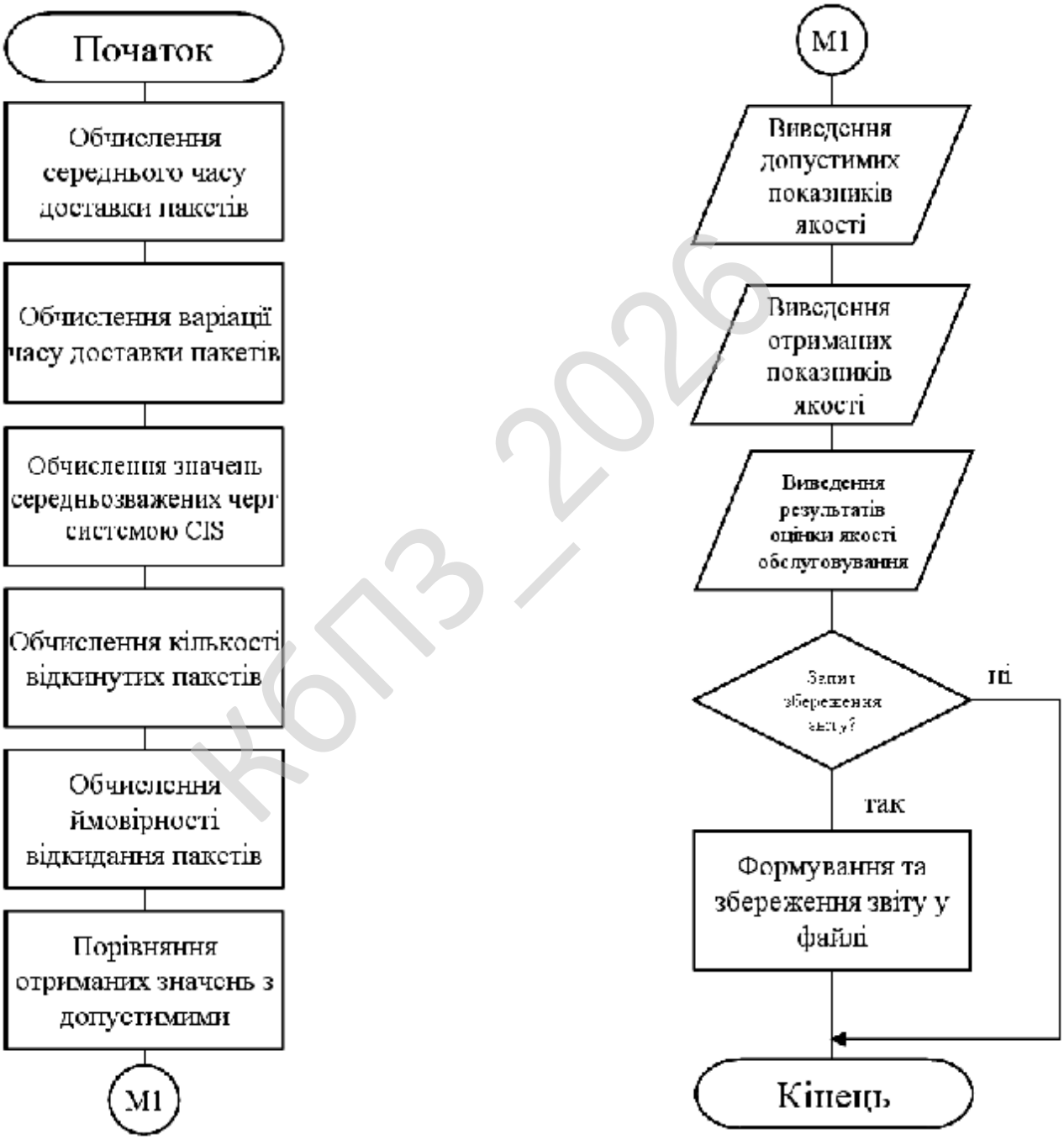


Рисунок 4.2 – Блок-схема роботи підпрограми

КОПЗ_2026

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

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

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

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

СКБД третього покоління повинні включити в себе СКБД другого покоління. Системи другого покоління внесли вирішальний вклад у двох областях – непроцедурний доступ за допомогою мови запитів SQL і незалежність даних. Ці досягнення обов'язково повинні враховуватися в системах третього покоління.

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

КПЗ_2026

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

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

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

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

Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо. Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповняться численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.

КПЗ – 2026

Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно ми спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової оперативної пам'яті. Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв. Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом – з паралельними обчисленнями і хмарними технологіями.

Основні характеристики СКБД

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

Можливості СКБД

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

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

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

КПІЗ_2026

Основні компоненти середовища СКБД

1. Апаратне забезпечення.

2. Програмне забезпечення.

3. Дані.

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

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

Архітектура СКБД

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

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.

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

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

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

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

Microsoft SQL Server – комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів – Transact-SQL, створена спільно Microsoft та Sybase.

Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за

КПЗ – 2026

розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

Походження

Базовий код MS SQL Server (до версії 7.0) ґрунтувався на коді Sybase SQL Server. Це дозволило Microsoft вийти на ринок баз даних для підприємств, де конкурували Oracle, IBM, і, пізніше, сама Sybase. Microsoft, Sybase і Ashton-Tate спочатку об'єдналися для створення і випуску на ринок першої версії програми, що отримала назву SQL Server 1.0 для OS/2 (близько 1989 року), яка фактично була еквівалентом Sybase SQL Server 3.0 для Unix, VMS та ін. Microsoft SQL Server 4.2 був випущений у 1992 році та входив до складу операційної системи Microsoft OS/2 версії 1.3.

Офіційний реліз Microsoft SQL Server версії 4.21 для ОС Windows NT відбувся одночасно з релізом самої Windows NT (версії 3.1). Microsoft SQL Server 6.0 був першою версією SQL Server, створеною виключно для архітектури NT і без участі в процесі розробки Sybase.

До того часу, як вийшла на ринок ОС Windows NT, Sybase і Microsoft розійшлися та створювали вже власні моделі цього програмного продукту. Microsoft намагалася отримати виняткові права на всі версії SQL Server для Windows. Пізніше Sybase змінила назву свого продукту на Adaptive Server Enterprise щоб уникнути плутанини з Microsoft SQL Server. До 1994 року Microsoft отримала від Sybase три повідомлення про авторські права як натяк на походження Microsoft SQL Server.

Після розділення компанії зробили декілька самостійних релізів програм. SQL Server 7.0 був першим сервером баз даних зі справжнім графічним інтерфейсом адміністрування. Для усунення претензій з боку Sybase у порушенні авторських прав, весь успадкований код в сьомій версії був переписаний. Це забезпечило також й успіх SQL Server 2000, який був першою редакцією, орієнтованою на архітектуру IA-64.

КПЗ – 2026

Протягом подальших шести років корпорація Microsoft працювала над вдосконаленням вже існуючої версії SQL Server 2000 доки не збудувала зручнішу систему Microsoft SQL Server 2005. Були вдосконалені продуктивність, кліентські інструменти інтегрованого середовища розробки, а також у декількох додаткових системах, що встановлюються разом із SQL Server 2005. Змінено: інструментарій процесів керування сховищами даних (SQL Server Integration Services або SSIS), сервер звітів, сервер OLAP та інтелектуального аналізу даних (Analysis Services), а також декілька технологій повідомлень, особливо Service Broker та Notification Services.

SQL Server 2025+

SQL Server 2025 включає низку вдосконалень для роботи з критичними бізнес-застосунками і бізнес-аналітикою як в традиційних дата-центрах, так і в приватних, публічних і гібридних хмарних середовищах. Серед нових можливостей SQL Server 2025 виділяються SQL Server AlwaysOn (рішення підтримки високого рівня доступності даних та аварійного відновлення), xVelocity (технологія збільшення продуктивності сховищ даних та програм бізнес-аналітики), нові рішення в області візуалізації PowerPivot і PowerView для створення звітів і аналітичних програм з Excel і SharePoint, покращені інструменти для інтеграції даних і управління ними, включаючи SQL Server Data Quality Services і Master Data Services, нова семантична модель бізнес-аналітики та інструмент для адміністраторів баз даних і розробників застосунків SQL Server Data Tools.

Також Microsoft зробила значні інвестиції в області Big Data, а саме в інтеграцію SQL Server і популярних інструментів для бізнес-аналітики з неструктурованою інформацією.

Microsoft робить SQL Server доступним у різноманітних варіантах, які різняться наборами властивостей в залежності від цілей кінцевого користувача. Це такі редакції як:

КПЗ-2026

SQL Server Compact Edition (SQL CE) – Компактний механізм бази даних. Завдяки малому обсягу (2 Мб для DLL) має зменшені властивості у порівнянні з іншими варіантами. Розмір бази даних обмежений 4 Гб і не може використовуватися як служба Windows.

SQL Server Express Edition – Раніше відомий під назвою MSDE (Microsoft SQL Server Desktop Engine), Microsoft SQL Server Express є вільно поширюваною версією SQL Server. Дана версія має деякі технічні обмеження, також відсутні графічні інструменти адміністрування. Такі обмеження роблять її непридатною для розгортання великих баз даних. В основному вона використовується у застосунках, при проектуванні, або для самостійного вивчення. Розмір бази даних обмежений 4 Гб, розмір пам'яті, що може бути адресованою–1 Гб, підтримує лише один процесор.

SQL Server Workgroup Edition – SQL Server Workgroup Edition включає функціональність ядра бази даних, але не включає додаткові сервіси.

SQL Server Standard Edition – SQL Server Standard edition включає механізми ядра бази даних, а також автономні сервіси. Відрізняється від варіанту Enterprise Edition тим, що підтримує менше активних вузлів та не включає деякі функції збільшення продуктивності.

SQL Server Enterprise Edition – SQL Server Enterprise Edition–це повнофункціональна версія SQL Server

SQL Server Developer Edition – SQL Server Developer Edition включає ті самі функції, що й SQL Server Enterprise Edition, але містить обмеження щодо використання його лише для розробки та тестування. Його ліцензія не дозволяє використання як виробничого сервера.

Функціональність

Microsoft SQL Server як мову запитів використовує версію SQL, що отримала назву Transact-SQL (скорочено T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє

КПЗ – 2026

використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія бази даних з керуючим застосунком).

Microsoft SQL Server та Sybase ASE для взаємодії з мережею використовують протокол рівня застосунка під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server також підтримує Open Database Connectivity (ODBC)– інтерфейс взаємодії застосунків з СУБД. Версія SQL Server 2005 надає можливість підключення користувачів через веб-сервер-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кроссплатформенно з'єднуватися з SQL Server. Microsoft також випустила сертифікований драйвер JDBC, що дозволяє застосункам під керування Java (таким як BEA і IBM Websphere) з'єднуватися з Microsoft SQL Server 2000 і 2005.

SQL Server підтримує дзеркалювання та кластеризацію баз даних. Кластер серверу SQL–це сукупність однаково конфігурованих серверів; така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному з серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.

SQL Server підтримує надлишкове дублювання даних за трьома сценаріями:

1. Знімок: Виконується «знімок» бази даних, який сервер відправляє одержувачам.

2. Історія змін: Всі зміни бази даних безперервно передаються користувачам.

3. Синхронізація з іншими серверами: Бази даних декількох серверів синхронізуються між собою. Зміни усіх баз даних відбуваються незалежно на

КБПЗ_2026

кожному сервері, а під час синхронізації відбувається звірка даних. Дублювання такого типу передбачає можливість вирішення протиріч між базами даних.

SQL Server 2005 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм RC5, який являє собою блоковий шифр із безліччю параметрів: розміром блоку, розміром ключа й числом раундів. В алгоритмі RC5 передбачені три операції: XOR, додавання й циклічні зрушення.

На більшості процесорів операції циклічного зрушення виконуються за постійний час, змінні циклічні зрушення являють собою нелінійну функцію. Циклічні зрушення залежать як від ключа, так і від даних.

В RC5 використовується блок змінної довжини, але в приводиться прикладі, що буде розглянутий, 64-бітовий блок даних. Шифрування використовує 2r+2 залежних від ключа 32-бітових слів – S_0, S_1, S_2,... S_{2r+1} – де r – число раундів. Для шифрування спочатку потрібно розділити блок відкритого тексту на два 32-бітових слова: A и B. (При впакуванні байтів у слова в алгоритмі RC5 дотримується угода про прямий порядок (little-endian) байтів: перший байт займає молодші біти регістра A й т. ін.) Потім:

$$A=A + S_0$$

$$B = B + S_0$$

КОПЗ_2026

Для i від 1 до r:

$$A = ((A \oplus B) <<< B) + S_{2i}$$

$$B = ((B \oplus A) <<< A) + S_{2i+1}$$

Вихід перебуває в регістрах A и B.

Розшифрування теж нескладно. Потрібно розбити блок відкритого тексту на два слова, A й B, а потім:

Для i від r до 1 із кроком -1:

$$B = ((B - S_{2i+1}) >>> A) \oplus A$$

$$A = ((A - S_{2i}) >>> B) \oplus B$$

$$B = B - S_i$$

$$A = A - S_0$$

Символом «>>>» позначене циклічне зрушення вправо**.** Звичайно ж**,** всі додавання й вирахування виконуються по модулю 2^{32}.

Створення масиву ключів складніше, але теж прямолінійно. Спочатку байти ключа копіюються в масив L із 32-бітових слів, доповнюючи при необхідності заключне слово нулями. Потім масив S ініціалізується за допомогою лінійного конгруентного генератора по модулю 2^{32}:

$$S_0 = P$$

Для i від 1 до $2(r + 1) - 1$:

$$S_i = (S_{i-1} + Q) \bmod 2^{32}$$

де P = 0xb7e15163 і Q = 0x9e3779b9.

Нарешті, потрібно підставити L в S:

$$i = j = 0$$

$$A = B = 0$$

Виконати 3n раз (де n – максимум від $2(r + 1)$ і c):

$$A = S_i = (S_i + A + B) <<< 3$$

$$B = L_i = (L_i + A + B) <<< (A + B)$$

$$i = (i + 1) \bmod 2(r + 1)$$

$$i = (j + 1) \bmod c$$

КПЗ_2026

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціо нальні розділи: Функціональних кнопок ПЗ; Верхнього меню; Розділу обрання групи; Розділу виведення результату роботи системи; Функції представлені у графічному вигляді.

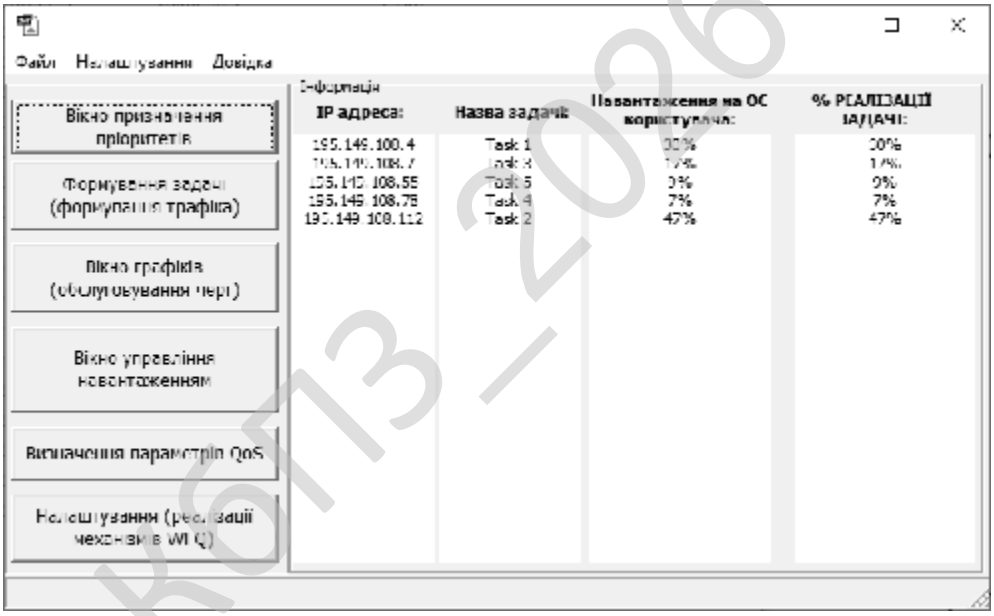


Рисунок 5.1 – Головне вікно ПЗ

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

КаПЗ_2026

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

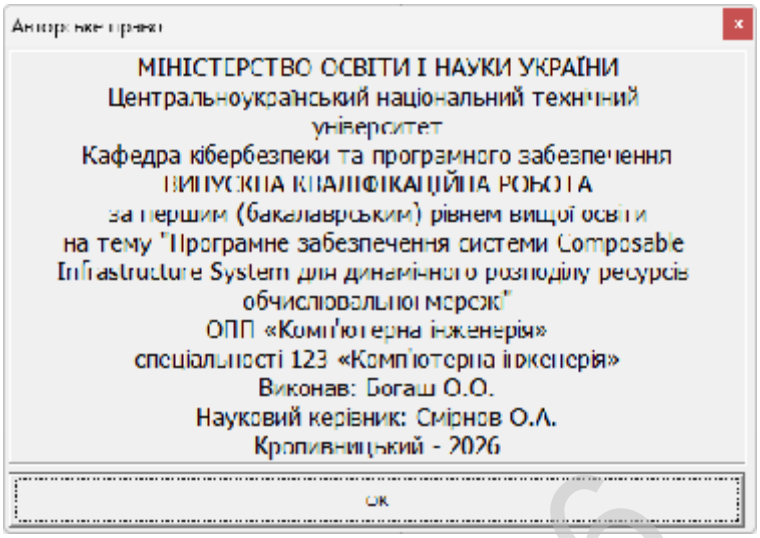


Рисунок 5.2 – Авторське право

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

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

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

КОПЗ_2026

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

КПЗ_2026

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

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

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

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

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

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

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

КПЗ_2026

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

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

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

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

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

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

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

КПЗ–2026

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт.

Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення.

Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

КбПЗ_2026

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– Досліджена система Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

– На основі отриманих результатів досліджень створена програмна реалізація системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі.

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

КПЗ_2026

системи Composable Infrastructure System для динамічного розподілу ресурсів обчислювальної мережі. 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 RC5.

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.

КбПЗ-2026

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

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

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

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

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

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

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

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

9. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

КОПЗ_2026

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

11. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

12. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

14. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

15. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

16. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.

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

КБПЗ_2026

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

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

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

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

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

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

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

КБПЗ-2026

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

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

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

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

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

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

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

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

КОПЗ_2026

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

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

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

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

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

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

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

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

КОПІЯ 2026

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

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

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

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

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

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

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

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

КБПЗ-2026

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

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

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

49. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дрєєв 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.

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

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

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

53. Смірнов О.А., Смірнов С.А., Дідик А.К., Дрєєв 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.

КБПЗ_2026

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

55. Смірнов О.А., Смірнов С.А., Дідик А.К., Дрєєв 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.

56. 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.

57. 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.

58. Смірнов О.А., Кавун С.В., Коваленко О.В., Дрєєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

59. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дрєєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

60. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

61. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

62. Смірнов О.А., Євсеєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. Навчальний посібник. – Кіровоград: КНТУ 2012. – 454 с

КОПЗ_2026