Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації”

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Волков С.В.
«____» _____ 2026 р.

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

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*

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

Волкову Сергію Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації*

2. Керівник роботи *Коваленко Олександр Володимирович, докт. техн. наук, професор*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 133-02 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 розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації*

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

1. Призначення та область використання.

2.Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника
_____ Коваленко О.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача
_____ Волков С.В.
(прізвище та ініціали)

АНОТАЦІЯ

Волков С.В. Програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації. 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 розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

Метою розробки є програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

Результат роботи – програмна реалізація системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

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

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

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

Програму розроблено в середовищі Visual C#.

Ключові слова: комп'ютерна інженерія, центр обробки даних, кастомізація

ABSTRACT

Volkov S.V. Software for the development of an individual data center project using customization technologies. 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, which is intended for the development of an individual data center project using customization technologies.

The purpose of the development is software for the development of an individual data center project using customization technologies.

The result of the work is the software implementation of the development of an individual data center project using customization technologies.

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 was developed in Visual C#.

Keywords: computer engineering, data center, customization

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. Щільність навантаження, розподіл повітря, розташування плитки для підлоги; проектування центру обробки даних зараз складніше, ніж будь-коли, але з урахуванням деяких передових практик створення ефективного та надійного проекту центру обробки даних вам під силу.

Під час проектування вашого центру обробки даних враховуйте початкові та майбутні навантаження, зокрема умови часткового та низького навантаження.

Зменште споживання енергії центром обробки даних та підвищте ефективність охолодження, групуючи обладнання з подібною щільністю теплового навантаження та температурними вимогами. Це дозволяє керувати системами охолодження до найменш енергоємних заданих значень для кожного місця розташування.

Визначте клас вашого центру обробки даних, щоб визначити рекомендовані та допустимі межі середовища:

– «Рекомендований» поєднує енергоефективну роботу з високою надійністю.

– «Допустиме» окреслює межі, перевірені виробниками IT-обладнання на функціональність.

Майте на увазі, що робота поза рекомендованим діапазоном може призвести до того, що вентилятори сервера працюватимуть на вищих швидкостях і, отже, споживатимуть більше енергії. Вищі температури зворотного повітря подовжують години роботи повітряних економайзерів. Вища температура рециркуляційного повітря підвищує ефективність охолоджувальної інфраструктури, заощаджуючи як енергію, так і гроші.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

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

– Огляд існуючих систем розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

– Дослідження системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

– Програмна реалізація системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Як уже було відзначено, спостереження про формування тенденції використання індивідуалізованих продуктів було озвучено на зустрічі виробників СКС, ринку не настільки динамічного, якщо не сказати інертного. Довгий час на ньому ніяких значимих змін не відбувалося, і тільки в останні два роки спостерігається підвищена активність, переважно пов'язана з появою нових стандартів, технологій передачі даних і виводом на ринок більше продуктивних компонентів (волокно OM5, мідні лінії Категорії 8).

Уже після зустрічі, проаналізувавши деякі проекти й випадки, коли була потрібна кастомізація (не тільки в частині СКС, але й в інших підсистемах), ми прийшли до виводу, що дана послуга найбільш затребувана саме на ринку ЦОДів, причому найбільшою мірою в корпоративних центрах обробки даних. Пояснення тому досить просте.

Власники комерційних ЦОДів найчастіше не мають повного подання про те, яке саме обчислювальне встаткування захочуть установити замовники і які будуть пред'являтися вимоги з забезпечення резервування живлення, швидкостям підключення та ін. У силу цих обставин експлуатуючі служби комерційних ЦОДів воліють застосовувати універсальні й модульні рішення.

Власники корпоративних ЦОДів і особливо так званих мегаЦОДів завжди намагаються створити максимально ефективну комплексну систему із середнім строком життя 10-15 років. У результаті вони прагнуть домогтися максимальної ефективності в умовах обмеженості ресурсів, з урахуванням особливостей геометрії приміщень і інших факторів, що лімітують. Саме тому й відбувається перехід від створення індивідуальних проектних рішень на базі штатних універсальних продуктів до підготовки повністю індивідуалізованого рішення:

індивідуальний підхід у проектуванні, доповнений кастомізованою продукцією. Як говорилося раніше, великі гравці IT-ринку можуть дозволити собі створення абсолютно нового, інноваційного, а часом навіть революційного рішення, у той час як у більшості власників корпоративних обчислювальних центрів таких можливостей немає. Однак, оскільки відповідна потреба є, деякі виробничі компанії орієнтуються у своєму бізнесі саме на приватні проекти й розробку унікальних продуктів під завдання або проект.

Наприклад, у нашої компанії є певний досвід створення унікальних монтажних конструктивів. Так, у рамках реалізації проекту для одного з комерційних ЦОДів передбачається поставити 1400 стійок, розроблених під замовлення (за станом на липень 2017 року поставлене й уведено в промислову експлуатацію 340 серверних і 20 кросових стійок). Даний проект є виключенням, що, як відомо, підтверджує загальне правило про те, що кастомізовані рішення найчастіше використовуються в корпоративних ЦОДах.

Мета проекту – створення надійного конструктиву з високим ступенем універсальності, сумісного з будь-яким установлюваним у нього встаткуванням клієнта. Саме тут перебуває одна із крапок демаркації встаткування власника й орендаря ЦОДу. У цей момент роботи у кастомізації тривають, і реалізуються два проекти, для одного й з яких уже поставляється техніка, а іншого перебуває на завершальній стадії остаточного узгодження прототипу.

1.2 Область застосування

Підходи до створення індивідуального продукту або рішення можна розділити на приватний і комплексний. У першому випадку передбачається об'єднання двох підсистем. Яскравим прикладом такого підходу є розробка унікального коннектора MXC компаніями US Conec і Intel. Основним завданням проекту було створення інноваційного з'єднувача для здійснення передачі даних на більше високих швидкостях. Відповідно до прийнятого поділу ролей компанія

US Conec розробляла пасивну частину лінії передачі даних (коннектор), у той час як Intel займалася активною складовою (кремнієвою фотонікою). Отриманий у результаті The Big Data Connector здатний забезпечити передачу даних на швидкостях до 1,6 Тбіт/с.

Комплексний підхід істотно відрізняється від частки. Топологічно відносини між продуктом кастомізації й пов'язаними з ним підсистемами нагадують зірку. Причому ці зв'язки найчастіше мають однобічний характер і спрямовані від підсистем до продукту, тобто задають вимоги для його створення або зміни. Як приклад можна розглядати підходи таких глобальних компаній, як Google і Facebook, які створюють для себе не просто унікальні продукти, а комплексні рішення й системи. Зокрема, компанія Марка Цукерберга в цей час веде розробку серверної шафи для своїх ЦОДів разом з Delta Electronics. У рамках даного проекту мова йде про зміну не стільки конструктиву, скільки концепції системи електропостачання ЦОДу, з одного боку, і про її узгодження з активним і серверним устаткуванням, з іншої.

Комплексний підхід оптимальний у тому випадку, коли розробляється високотехнологічний продукт, взаємодіючий з великою кількістю підсистем. Приватний же підхід застосуємо там, де необхідно забезпечити сполучення двох підсистем і створити або змінити для цього певне рішення.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

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

Fujitsu Data Center Management & Automation: рішення

Data Center Management & Automation містить у собі рішення для автоматизації процесів, керування рівнем обслуговування – керування якістю обслуговування й керування електроживленням у центрі обробки даних.

Автоматизація процесів

Для автоматизації процесів і планованого підвищення ефективності базову технологію необхідно масштабувати відповідно до оновлених процесів. Після цього процеси варто інтегрувати в уніфіковані робочі процедури, що дозволить центрам обробки даних надавати комплексні IT-послуги замість IT-систем. Шляхом максимально ефективного впровадження процесів, сумісних з ITIL® (ITIL® is a Registered Trade Mark of AXELOS Limited), IT-менеджери можуть підвищити ефективність центрів обробки даних. Автоматизація процесів дозволяє одержати досвід ефективного керування й обробки повторюваних процесів. Сюди входять процедури планованих сценаріїв для визначення ефективного способу дії.

Керування рівнем обслуговування – керування якістю обслуговування

Керування рівнем обслуговування (SLM) дозволяє підтримувати стабільну якість обслуговування за рахунок безперервного моніторингу якості обслуговування центра обробки даних і автоматизації процесів замовника по залученню людей, інтеграції IT-процесів і технологій у ланцюжок надання IT-послуг. Разом з технологіями CA наших партнерів ми надаємо комплексний стек

рішень зі стандартними процедурами, інструкціями й рекомендаціями для швидкої реалізації.

Керування електроживленням у центрі обробки даних

Центри обробки даних споживають основну частину електроенергії в організації. Не дивно, що у зв'язку з швидкорастущими витратами на електроенергію організації шукають ефективні способи моніторингу ефективності витрати електроенергії в центрах обробки даних і IT-середовища. Компанія Fujitsu пропонує рішення з моніторингу споживання електроенергії в центрі обробки даних і надає звіти по серверах і системам зберігання, які допомагають визначити, яке встаткування споживає найбільше електроенергії. Дані, передані на інтуїтивно зрозумілу інформаційну панель, допомагають організаціям оптимізувати використання IT-ресурсів і знизити витрати.

Керування інформаційними технологіями як послуга

Розроблена компанією Fujitsu модель керування інформаційними технологіями як послуги (ITMaaS) – це пакет застосунків на основі SaaS, що забезпечує інфраструктуру, поточний контроль роботи застосунків і набір функціональних можливостей довідкових служб, необхідні для керування інформаційними технологіями з мінімальними витратами сил і засобів.

У цей час у пакет включено два модулі на основі розробленою компанією Nimsoft передової технології керування послугами.

– Служба підтримки як послуга. Комплексна система керування інформаційно-технологічними службами з убудованими методиками з бібліотеки ITIL® і покроковими технологічними процесами, що забезпечує погоджене оптимізоване керування всіма аспектами надання послуг.

– Поточний контроль як послуга. Єдиний уніфікований інтерфейс відстеження всіх найважливіших ресурсів підприємства в їхньому динаміку – як засобами хмарних обчислень, так і в локальному режимі. Він надає всі необхідні засоби комплексного поточного контролю для постійної роботи служб підприємства на оптимальному рівні продуктивності.

Fujitsu ManageNow®

Fujitsu ManageNow® – це набір модульних рішень компанії Fujitsu для простого автоматизованого впровадження й ефективного використання системи керування IT-середовищем підприємства. Рішення ManageNow® розроблені для керування гетерогенними IT-середовищами. Рішення ManageNow® засновані на точно вивірених наборах програмного забезпечення й послуг: ManageNow® містить у собі продукти для керування IT-середовищем підприємства, доповнення й розширення, перевірені на практиці концепції й рішення, а також використання методології Fujitsu на всіх стадіях проекту. Наш підхід містить у собі послуги по інтеграції, обслуговування й підтримку.

ManageNow® для моніторингу ЦОД

ManageNow® для моніторингу центра обробки даних – це високомасштабуєме рішення для керування гетерогенними IT-інфраструктурами. Рішення ManageNow® для моніторингу центра обробки даних підтримує консолідацію різних продуктів в області моніторингу й керування, у тому числі програмного забезпечення по керуванню з відкритим вихідним кодом, у вигляді єдиної панелі керування. Рішення ManageNow® для моніторингу центра обробки даних уніфікує й централізує процеси моніторингу й надає організаціям всебічне подання IT-інфраструктури на основі ролей. Рішення автоматично ідентифікує проблеми, класифікує й призначає їм пріоритет на основі потенційно можливого впливу на підприємство, а також ініціює процеси корекції помилок на основі правил. Графічний користувальницький інтерфейс рішення ManageNow® для моніторингу центра обробки даних надає адміністраторам зручні візуальні подання даних, тим самим зменшуючи складність керованої інфраструктури. Попередньо зконфігуроване керування подіями й призначення об'єктів, що відслідковуються, бізнес-процесам спрощує керування IT-середовищем підприємства. Цикл автоматизованого контролю, реалізований рішенням Fujitsu ManageNow® для моніторингу центра обробки даних, оптимізує щоденні операції центра обробки даних.

Рішення ManageNow® для керування клієнт-серверною інфраструктурою

Рішення ManageNow® для керування клієнт-серверною інфраструктурою спрощує адміністрування й моніторинг неоднорідної інфраструктури серверів, ПК і ноутбуків і значно оптимізує роботу. Спектр його функцій досить широка: інвентаризація апаратного й програмного забезпечення, створення програмних пакетів, у тому числі стандартних пакетів Microsoft MSI, каскадний розподіл програмного забезпечення, автоматична установка операційної системи, керування виправленнями, віддалене обслуговування й повне керування системами на робочих місцях. ManageNow® Client & Server Management дозволяє збільшити продуктивність і ефективність роботи за чіт автоматичної установки й керування стандартизованими системами. Стандартизація й автоматизація керування життєвим циклом у клієнт-серверних інфраструктурах дає замовникам цілий ряд переваг для ефективного й економічного керування постійно, що ускладнюється взаємодією, між апаратними компонентами й програмними продуктами.

Рішення ManageNow® для роздрібної торгівлі

Потреби в інвентаризації апаратного й програмного забезпечення, створення пакетів програмного забезпечення, каскадне поширення програмного забезпечення, автоматична установка операційної системи, керування виправленнями, дистанційне обслуговування, повне всебічне керування робочим місцем і системами роздрібної торгівлі є одними з найбільш важливих завдань для IT-інфраструктур роздрібної торгівлі. Рішення ManageNow® для роздрібної торгівлі дозволяє впоратися із завданнями по автоматичній установці й налаштуванню стандартних продуктів за допомогою автоматичної відтвореної клієнт-серверної установки. Рішення ManageNow® для роздрібної торгівлі дозволяє робити модифікації, що набудовуються під вимоги замовника, і розширення, надає базу даних автоматично конфігуруємих драйверів, керування активами й віддалений контроль IT-інфраструктури.

Програмно-визначаємий ЦОД Windows Server

Програмний центр обробки даних (SDDC) – це розповсюджений термін, що звичайно відноситься до центра обробки даних, де вся інфраструктура віртуалізована. Віртуалізація відіграє ключову роль. Це означає, що встаткування й програмне забезпечення в ЦОД виходять за рамки традиційного співвідношення "один до одному". За рахунок емуляції встаткування за допомогою гіпервізора операційні системи й додатки можна абстрагувати від фізичного встаткування й помножити для об'єднання процесорів, пам'яті, пристроїв уведення-виводу й мереж в еластичні пули ресурсів.

Реалізація програмного ЦОД корпорації Майкрософт складається з технологій Windows Server, описуваних у цій статті. У її основі лежить гіпервізор Hyper-V, що надають платформу віртуалізації, на якій формуються інфраструктура мережі й сховище. Технології безпеки, розроблені для унікальних завдань віртуалізованої інфраструктури, нейтралізують внутрішні й зовнішні погрози. Завдяки убудованій в Windows Server оболонці PowerShell у сполученні з System Center і/або Operations Management Suite можна програмувати й автоматизувати підготовку, розгортання, налаштування й керування.

Технології в складі Windows Server і System Center є основними компонентами програмного ЦОД Windows Server. Але незважаючи на те, що це віртуалізована платформа, у її основі повинне лежати відповідне встаткування. Партнери корпорації Майкрософт, що беруть участь у рішеннях Windows Server Software-Defined (WSSD), і програми Azure Stack HCI Solutions можуть допомогти вашому корпоративному придбанню підходящого встаткування і його запуску в нуль-й день.

Azure Stack рішення HCI

Створення програмного центра обробки даних на основі Windows Server із правильною інфраструктурою встаткування є найважливішим етапом успіху. Саме тому ми зібралися з 15 партнерами по створенню проектів SDDC, перевірених корпорацією Майкрософт, і рекомендацій з розгортання.

Партнери корпорації Майкрософт пропонують набір рішень, які працюють із Window Server 2019 через програму Azure Stack HCI й Windows Server 2016 за допомогою програми Windows Server Software-Defined (WSSD) для забезпечення високопродуктивної, Ний і мережної пам'яті, зберігання й мережі. Infrastructure. Гіперконвергентні рішення поєднують ресурси обчислень, зберігання й мережі у відповідним галузевим стандартам серверах і компонентах для поліпшеної аналитики й контролю в ЦОД.

Віртуалізовані технології Windows Server

У частині, що залишилася, цього розділу перераховані технології програмного ЦОД Windows Server і наведені посилання на документацію по кожнійій з них.

Windows Server, технологія Hyper-сходження

Технології віртуалізації Windows Server містять у собі відновлення для Hyper-V, віртуального комутатора Hyper-V, а також для захищеної структури й екранованих віртуальних машин. Ці відновлення підвищують безпека, масштабованість і надійність. Завдяки відновленням відказостійкої кластеризації, мережних компонентів і сховища ці технології ще зручніше розгортати й адмініструвати при використанні в сполученні з Hyper-V.

Гіпервізор Hyper-V

Hyper-V – це технологія віртуалізації на основі гіпервізора для Windows. Гіпервізор ключовим компонентом технології віртуалізації. Це процесор-залежна платформа віртуалізації, що дозволяє декільком ізольованим операційним системам використовувати загальну апаратну платформу.

Кластеризація гостюючи із загальним VHDX

Загальний диск VHDX відрізняється гнучкістю й безпекою, а також не прив'язаний до топології базового сховища, тому він усуває необхідність надавати фізичне базове сховище для гостьовий ОС. Новий загальний диск VHDX підтримує оперативна зміна розміру.

– Загальний диск VHDX може розташовуватися в загальному томі кластера (CSV) у блоковому сховищі або у файловому сховищі SMB.

– Загальні VHDX-файли підтримують репліку Hyper-V і резервне копіювання на рівні вузла.

Убудована програмна реплікація віртуальних машин у мережі з використанням сертифікатів. Відсутність прив'язки до серверів, мережному встаткуванню або встаткуванню для зберігання даних в обох середовищах.

Відсутність потреби в інших технологіях реплікації віртуальних машин і, як наслідок, зниження витрат.

– Автоматичне виконання динамічної міграції.

– Зручність налаштування й керування через диспетчер Hyper-V, PowerShell або Azure Site Recovery.

Мережний контролер

Централізована програмувальна крапку автоматизації для керування, налаштування, моніторингу, а також виявлення й усунення несправностей у віртуальній і фізичній мережній інфраструктурі центра даних.

Адміністратори використовують засіб керування, взаємодіюче прямо з мережним контролером. Мережний контролер надає засобу керування відомості про мережну інфраструктуру, включаючи віртуальну й фізичну інфраструктуру.

Брандмауер центра обробки даних

Якщо цей компонент розгорнуть і надається як послуга, адміністратори клієнта можуть установити й настроїти політики брандмауера, щоб захистити віртуальні мережі від небажаного трафіку з Інтернету й інтрамереж.

Адміністратор постачальника послуг або адміністратор клієнта можуть управляти політиками брандмауера ЦОД через мережний контролер.

Об'єднання впроваджених комутаторів

SET є альтернативним рішенням для об'єднання мережних карт, яке можна використовувати в середовищах з Hyper-V і стеком програмно-конфігуруємої мережі (SDN).

Програмне балансування навантаження

SLB дозволяє розміщати те саме робоче навантаження на декількох серверах, що забезпечує високий рівень доступності й масштабування. Можливості масштабованого балансування навантаження з використанням віртуальних машин SLB на тих же серверах Hyper-V, що використовуються для інших робочих навантажень віртуальних машин. SLB підтримує швидке створення й видалення кінцевих крапок балансування навантаження для операцій постачальників хмарних послуг. SLB підтримує десятки гігабайтів на кластер, надає просту модель підготовки й зручна в масштабуванні.

Локальні дискові простори

Локальні дискові простори використовують стандартні сервери з локально підключеними дисками для створення програмно-визначаємого сховища з високим рівнем доступності й масштабованості, причому вартість такого сховища значно нижче вартості традиційних масивів SAN або NAS. Архітектура такого сховища істотно спрощує підготовку ресурсів і розгортання.

Локальні дискові простори представляють нову шину Software Storage Bus, а також у них використовується безліч існуючих можливостей Windows Server, таких як відказостійка кластеризація, загальні томи кластера (CSV), протокол SMB3 і, звичайно, дискові простори.

Якість обслуговування сховища

Централізовані відстеження й контроль продуктивності сховища віртуальних машин за допомогою ролей Hyper-V і масштабованого файлового сервера допомагають забезпечити рівномірне використання ресурсів сховища між декількома віртуальними машинами. Служба якості обслуговування сховища убудована в програмно-визначаєме рішення Майкрософт для зберігання даних, надаваному масштабованим файловим сервером і Hyper-V з використанням протоколу SMB3.Новий диспетчер політик забезпечує централізований моніторинг продуктивності сховища.

Репліка сховища

Аварійне відновлення й готовність до аварійних ситуацій допомагають повністю виключити втрати даних, а також дозволяють організувати синхронний захист даних, розташованих у різних стійках, на різних поверхах, у різних будинках, філіях, містах і країнах за рахунок більше ефективного використання декількох центрів обробки даних.

Синхронна реплікація

– Додаток записує дані.

– Дані журналу записуються й реплікуються на віддалений сайт.

– Дані журналу записуються на віддалений сайт.

– Підтвердження від віддаленого сайту.

– Підтвердження операції запису застосунку.

Захищена структура

Постачальники хмарних послуг і адміністратори корпоративної приватної хмари можуть використовувати захищену структуру для створення більше безпечного середовища для віртуальних машин. Захищена структура складається з однієї служби захисту вузла (HGS), що є, як правило, кластером із трьох вузлів, одного або декількох захищених вузлів і набору екранованих віртуальних машин.

Екрановані віртуальні машини

Дані й стан екранованої віртуальної машини захищені від перевірки, крадіжки й несанкціонованих змін з боку як шкідливих програм, так і адміністраторів центра обробки даних:

– Екрановані віртуальні машини запускаються тільки в структурах, призначених власниками віртуальних машин.

– Екрановані віртуальні машини шифруються службою BitLocker або іншим способом, щоб їх могли запускати тільки призначені власники.

– Працюючі віртуальні машини можна перетворити в екрановані.

Служба захисту вузла

Служба захисту вузла містить ключі від припустимих структур, а також від зашифрованих віртуальних машин.

Атестація працездатності пристрою

Атестація допомагає підвищити рівень корпоративної безпеки шляхом відстеження й підтвердження захисту на апаратному рівні без істотного росту експлуатаційних витрат.

Показаний вище довірений режим устаткування забезпечує найвищий рівень вірогідності завдяки довірі на основі убудованого в устаткування модуля TPM версії 2.0 і відповідності політиці цілісності коду для випуску ключів.

Налаштування необхідного стану PowerShell

Налаштування необхідного стану Windows PowerShell – це убудована в Windows платформа керування конфігураціями, заснована на відкритих стандартах. DSC відрізняється достатньою гнучкістю для надійної, погодженої роботи на кожному етапі життєвого циклу розгортання (розробка, тестування, підготовка, виробництво), а також при масштабуванні.

DSC підтримує "безперервні розгортання", тому можна багаторазово розгортати конфігурації, нічого не порушуючи.

– Для прискорення розгортання в конфігураціях DSC застосовуються лише ті параметри, які змінилися в порівнянні з вихідним значенням.

– DSC можна використовувати в локальному, загальнодоступному або приватному хмарному середовищі.

– DSC можна інтегрувати з будь-яким рішенням Майкрософт або стороннього постачальника, якщо зберігається можливість виконання сценаріїв PowerShell у цільовій системі.

VMM System Center

Диспетчер віртуальних машин входить у пакет System Center, що використовується для налаштування, адміністрування й перетворення

традиційних ЦОД з метою централізованого керування локальними середовищами, постачальниками послуг і хмарою Azure.

– Datacenter Налаштування компонентів центра обробки даних і керування ними як єдина структура в VMM.

– Вузли віртуалізації: VMM може додавати, підготовляти й адмініструвати вузли й кластери віртуалізації Hyper-V і VMware.

– Мережі VMM забезпечує віртуалізацию мережі, включаючи підтримку створення віртуальних мереж і мережних шлюзів і керування ними.

– Сховище: VMM може виявляти, класифікувати, підготовляти, розподіляти й призначати локальне й віддалене сховище.

Windows Admin Center

Windows Admin Center являє собою набір, що розвертається локально, засобів керування на основі браузера, що дозволяє управляти розташованими на об'єкті серверами під керуванням Windows незалежно від Azure і хмари. Windows Admin Center надає IT-адміністраторам повний контроль над всіма аспектами серверної інфраструктури й особливо корисний для керування в приватних мережах, які не підключені до Інтернету.

Публікація веб-сервера на DNS і налаштування корпоративного брандмауера можуть дозволити вам одержати доступ до Windows Admin Center з Інтернету, що забезпечить можливість підключення й керування серверами з будь-якого місця за допомогою Microsoft Edge або Google Chrome.

Huawei

Багато хто, безсумнівно, знають Huawei не тільки як виробника смартфонів, але і як компанію-розроблювача величезного спектра IT- і телекомунікаційних продуктів, швидко посилюючої своєї позиції на українському ринку.

На світовому ринку компанія в останні роки посідає друге місце по темпах росту бізнесу серед виробників серверів і перше – серед постачальників систем зберігання даних.

Системами зберігання даних користуються понад 2000 замовників у більш ніж 100 країнах, а серверами – більше 5000 замовників з 50 країн. В одному із проектів розгорнуто більше 100 тис. серверів Huawei. Серверами компанії встановлено більше 100 рекордів по продуктивності.

Хмарні дата-центри й мега-ЦОДи

Центри обробки даних зараз – чи не єдиний швидко зростаючий сегмент українського телекомунікаційного ринку. 12 млрд грн. заробили в 2025 році власники комерційних дата-центрів. Це на 28% більше, ніж роком раніше. Кількість комерційних стійок у дата-центрах виросло на 28% – до 25 500.

Huawei далеко не новачок і на ринку ЦОДобудування. Починаючи з 2002 року, компанія реалізувала по усьому світі більше 480 проектів дата-центрів різного масштабу й рівня складності. З них 160 – це інноваційні площадки для надання хмарних послуг. Huawei співробітничає з більш ніж 500 замовниками, що використовують хмарні обчислення. Активно розвиваються новітні підходи до створення дата-центрів – программно-конфігуруємі мережі (SDN) і ЦОДи (SDDC).

Територія діяльності Huawei не обмежується тільки Китаєм: реалізовані проекти в Південно-східній Азії, Африці, Америці, Європі. Є проекти й в Україні.

Проекти реалізовані компанією Huawei у світі

Два великих ЦОДу побудовані в КНР для компанії China Unicom. Вартість першого проекту склала близько 1 млрд доларів. Це четвертий по величині ЦОД у світі. Його загальна площа – 610 тис. кв. м.

Другий ЦОД вартістю в 1,97 млрд доларів – п'ятий у світі по своїх масштабах. Його площа становить майже 600 тис. кв. м.

У деяких проектах застосовані цікаві запатентовані технічні рішення, які дозволили домогтися високої енергоефективності. Наприклад, у ЦОД, побудованому Huawei для компанії China Mobile середньорічний коефіцієнт PUE=1,22, а мінімальне середньомісячне значення PUE становить 1,12.

Рішення Huawei для центрів обробки даних можуть допомогти компаніям успішно розвивати свій бізнес в епоху хмарних обчислень завдяки швидкому розгортанню, гнучкому розширенню, ефективності й надійності, а також інтелектуальному керуванню. 45 з 50 операторів зв'язку у світі, що забезпечують взаємодію хмар, використовують рішення Huawei. Співробітничаючи з Huawei, замовник одержує закінчений ЦОД «під ключ» з одних рук з абсолютно зрозумілою системою гарантії й підтримки.

Поряд з будівництвом ЦОДу і його сертифікацією договір передбачає дворічну підтримку. Реалізуючи подібні проекти, виробник одержує можливість підготувати проект із урахуванням свого портфеля рішень, а крім того, вивчити на практиці технологічні потреби українського ринку дата-центрів, що дозволить точніше оцінювати попит на рішення й адаптувати лінійку продукції до вимог ринку.

ЦОД у контейнері

Модульні рішення компанії дозволяють поряд з мега-ЦОДами реалізовувати й рішення, які можуть задовольнити вимоги бізнесу будь-якого масштабу, починаючи із самого маленького. Одним з варіантів модульного підходу до побудови дата-центрів стали мобільні й контейнерні ЦОДи – компактні, автономні й швидко встановлювані, зручні для транспортування й монтажу. Ці рішення нового покоління для центрів обробки даних містять у собі повну інтеграцію стійок, систему електропостачання й розподілу живлення, систему охолодження, програмне забезпечення системи керування й багато чого іншого. Їхні переваги полягають у швидкості розгортання, можливості масштабування за рахунок установки додаткових контейнерів, економії витрат. Для монтажу такого «ЦОД у контейнері» досить рівної площадки, електрики й каналу зв'язку. Треба відзначити, що згідно даним незалежних досліджень, за результатами 2025 року Huawei і HP розділили перше місце на світовому ринку контейнерних ЦОДів: кожної з них належить 11% даного ринку.

Що ж пропонується компанією для створення ЦОД:

Контейнерний ЦОД – готова інфраструктура для розміщення серверів, систем зберігання даних і іншого IT-устаткування, що складає з контейнера або набору контейнерів з усіма компонентами дата-центра, тобто серверними стійками, комунікаціями, системами електроживлення, охолодження й т.п. Такий ЦОД можна перевозити транспортом і в стислий термін збирати на місці.

Модульні ЦОДи виробництва Huawei призначені як для установки в серверних залах, так і для створення дата-центрів у непідготовлених приміщеннях (наприклад, на складах).

Модульні ЦОДи, а також контейнерні рішення «усе в одному» або кластерні контейнерні ЦОДи дозволяють нарощувати потужність поетапно, орієнтуючись на потребі клієнта.

Розглянемо ці рішення небагато більш детально

Контейнерний ЦОД IDS 1000-A «усе в одному» являє собою повноцінне інфраструктурне рішення, що включає в себе систему розподілу живлення й охолодження, стійки з IT-устаткуванням, кабельну систему, систему пожежогасіння, аварійного висвітлення й моніторингу.

Контейнерний ЦОД IDS1000 створений для використання поза приміщеннями в різних кліматичних зонах і має знижене енергоспоживання. Стандартний робочий діапазон температур (-40С – +55С) може бути розширений залежно від умов місцевості, де буде встановлюватися ЦОД.

Модульний контейнерний ЦОД IDS 1000-3i складається із трьох типів контейнерів стандартних габаритів: один тип служить для розміщення увстаткування (IT-контейнер), в іншому монтуються системи електроживлення, а в третьому – системи охолодження. Їх може доповнювати контейнер для розміщення персоналу й керування. У такий спосіб замовник може зібрати ЦОД будь-якій необхідній ємності й потужності як з конструктора шляхом комбінації модулів різних типів. Контейнерне виконання означає заводське виготовлення й попереднє тестування, високий ступінь інтеграції, зручність при роботі на місці

установці, скорочення часу розгортання до 80%, енергоефективність (PUE до 1.5), зручність експлуатації й технічного обслуговування, а також скорочення експлуатаційних витрат до 30%.

Компактний модульний ЦОД серії IDS2000, подібно IDS1000 містить всі необхідні підсистеми. Його конструкція дозволяє встановлювати один або два ряди стійок, що підвищує ефективність використання площі й охолодження з використанням гарячих і холодних коридорів. Потужність, споживана кожною стійкою в такому ЦОДі, може досягати 21 кВт. Для чого все це робиться? Відповідь проста – заради кращої масштабованості, зниження CAPEX, зменшення PUE, швидкого розгортання на площадці й недорогого віддаленого керування.

ІБЖ

Швидкий розвиток хмарних технологій – це ще більш строгі вимоги до електроживлення серверів і комутаторів. Для надійного електропостачання IT-устаткування Huawei пропонує енергоефективні інтелектуальні ІБЖ. Вони дозволяють вирішувати завдання масштабування, забезпечення високої доступності, зниження вартості обслуговування. Серед власних розробок Huawei для побудови інженерної інфраструктури особливої уваги заслуговують високоефективні онлайнові ІБЖ із подвійним перетворенням потужністю. Лінійка ІБЖ Huawei UPS 8000-D, UPS 5000-A, UPS 5000-E і UPS 2000-G потужністю від 1 до 800 кВа розрахована на додатки різного масштабу. У лінійці Huawei представлені як моноблочні, так і модульні ІБЖ, причому в сегменті модульних ІБЖ компанія займає одне із провідних місць по щільності живлення на займану площу.

Системи охолодження

У лінійці продуктів Huawei є лінійки внутрірядних і шафових прецизійного кондиціонерів для охолодження дата-центрів. Вони відрізняються цікавими конструктивними рішеннями, які полегшують процес монтажу й експлуатації встаткування. Особливої уваги заслуговує внутрірядний

кондиціонер шириною всього лише 300 мм, холодопродуктивністю 20 КВт, з убудованим парозволожувачем: Крім подвійного уведення електроживлення передбачені можливості уведення трубопроводів зверху й знизу й вентилятори з функцією гарячої заміни. Всі кондиціонери штатно оснащуються контролерами з 7-дюймовими кольоровими сенсорними дисплеями. Вивід різноманітної інформації на дисплей помітно спрощує монтаж, пуско-налагодження, експлуатацію й обслуговування встаткування.

Крім подвійного уведення електроживлення передбачені можливості уведення трубопроводів зверху й знизу й вентилятори з функцією гарячої заміни.

Енергоеффективні ІБЖ у сполученні з кондиціонерами із системою фрікулінгу дозволяють знизити PUE до 1,25.

Система керування

Інтелектуальна система керування NetEco від Huawei допоможе одержувати інформацію про аварії й відмови або просто змінах параметрів всіх інженерних систем ЦОДу в режимі реального часу й попередить про те, що якийсь елемент системи перебуває в незадовільному стані й незабаром може вийти з ладу. Крім цього система NetEco у повному обсязі забезпечує моніторинг параметрів мікроклімату ЦОДу.

NetEco відображає архітектуру центра обробки даних і звіти для користувача. Вона застосовується в рішеннях для центрів обробки даних IDS1000 і IDS2000, підвищує доступність ЦОДу, забезпечує відстеження параметрів струму й навантаження на ІБЖ, оперативне виявлення несправності, віртуальний контроль, динамічне керування PUE і оптимізацію служб систем живлення й охолодження.

Цікаве рішення – технічне обслуговування з 3D-візуалізацією, тобто динамічним 3D-відображенням температури для контролю й усунення крапок перегріву. Устаткування нижнього рівня (системи живлення, моніторингу параметрів навколишнього середовища, системи відеоспостереження й контролю доступу) легко підключається до NetEco за допомогою ModBus, SNMP і

інтелектуального протоколу зв'язку встаткування. Можна настроїти в NetEco і підтримку нестандартних протоколів.

Система реалізована в декількох версіях, найпростіші з яких поставляються штатно з ІБЖ. Партнери й замовники можуть відвідати демо-центр Huawei і побачити систему керування в дії.

Як писалося вище, компанія робить і реалізує встаткування власної розробки. У Китаї, Німеччині, Україні й інших країнах в Huawei є власні центри досліджень і розробки. При необхідності локальні команди можуть використовувати досвід колег з інших країн або, навпаки, надати підтримку закордонним колегам, а також використовувати новітні рішення компанії й при необхідності швидко адаптувати її продукти під вимоги замовників. Власні високотехнологічні заводи, а так само 100% тестування продукції дозволяють забезпечити високу якість.

Вибираючи Huawei як постачальник устаткування або рішення, замовник у підсумку одержує не тільки високоякісне закінчене рішення, але й професійний консалтинг міжнародної команди фахівців, здатної вирішити будь-які поставлені завдання в короткий термін і з високою якістю.

Пропрієтарні вендори, на прикладі VMware

В VMware зібрали під єдиним брендом продукти на базі яких можна зібрати повноцінний програмно-визначаємий ЦОД, тобто SDDC. Плюс навколо VMware працює ціла екосистема технологічних партнерів, які розширюють можливості ПЗ.

Платформа віртуалізації vSphere на сьогоднішній день є фактично стандартом «де-факто» при побудові віртуальних інфраструктур в Enterprise-сегменті, з релізом 6-й версії (ледве менш року тому) одержала можливість переміщати віртуальні машини без зупинки навіть у рамках континенту (round trip не повинен перевищувати 100 мс). А для того, щоб організувати єдиний мережний простір прекрасно підійде продукт для віртуалізації мережі – VMware NSX. Який за суттю зараз при установці стає частиною платформи практично

непомітно для адміна. Для зберігання віртуальних машин тут є широкі можливості для вибору… В VMware «рідні» рішення:

– Virtual Volumes (https://www.vmware.com/products/vsphere/features/ virtual-volumes).

– Virtual SAN (https://www.vmware.com/products/ virtual-san/).

Про SDS зверніть увагу ще на продукти Atlantis, PernixData і DataCore.

Як сама платформа керування хмарою є лінійка vRealize. Це саме всі ті продукти які автоматизують, дають портал самообслуговування й білінг ресурсів.

Openstack

Опенстек – це опенсурс. Усі знають. Для приватної хмари це може бути вкрай корисно. Саме тому що врядчи ви побачите 2 однакові реалізації хмари. Просто тому що організації всі різні, свої процеси усередині й культура IT.

У підсумку є можливість кастомізації на дуже високому рівні. Важливо розуміти, що кастомізація досягається шляхом переписування або дописування програмного коду, відповідно.

Складові частини Openstack перераховані отут (http://openstack.ua/about/components/). Компоненти ідеологічно VMware все ті ж. Звичайно рівень зрілості кожного з них може відрізнятися. Але зараз уже навіть телекому оператори говорять, ми будемо використовувати це. А все специфічне ми підчепимо вже потім.

І якщо з першого погляду ймовірно здасться раз опенсурс, значить звичне корпоративно-ентерпразне вам прикрутити буде складно. А от і не так.

Є й рішення по SDN, наприклад, OpenDaylight або OpenContrail, що, до речі, розробляється за підтримкою Juniper, одного з основних гравців на ринку «традиційного» мережного встаткування.

Є SDS, наприклад Datacore має можливість роботи з Openstack, поєднуючи й транслюючи всі вже існуючі системи зберігання в зрозумілому для Openstack виді.

Допустимо, якщо вам потрібний гарний балансувальник, ви можете сміло брати F5. У нього є інтергація з нейтроном. Є ще цікавий стартап Avi networks.

Вони роблять і SDN, NFV, і балансувальники, так само для Openstack. Та й взагалі зараз гарним тоном для ентерпрайз софта стала інтеграція з Openstack.

Імовірно читач на цьому моменті може подумати, а от адже в нас опенсурс. Чому ми знову про платний софт. А це тому що найкраще розглядати Openstack як платформу саме хмари, то об'єднуюча ланка яке автоматизує й зв'язує певні функції, а не тільки надає їх саме. Оскільки саме автоматизація – його основне завдання.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення написане мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об’єктно-орієнтована мова, призначена для розробки різноманітних безпечних і потужних застосунків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські застосунки Windows, веб-служби XML, розподілені компоненти, застосунки типу “ сервер-клієнт”, застосунки баз даних і багато яких інших. В Visual C# є розширений редактор коду, конструктори зі зручним користувальницьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку застосунків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

Синтаксис Visual C# дуже виразний, але простий у вивченні. Усі, хто знаком з мовами C, C++ або Java з легкістю взнають синтаксис із фігурними дужками, характерний для мови Visual C#. Розроблювачі, що знають кожну із цих мов, як правило, зможуть домогтися ефективної роботи з мовою Visual C# за дуже короткий час. Синтаксис Visual C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вираження й прямий доступ до пам'яті, чого немає в Java. Visual C# підтримує універсальні методи й типи, забезпечуючи більше високий рівень безпеки й продуктивності, а також ітератори, що

дозволяють при реалізації колекцій класів визначати власне поводження ітерації, що може легко використовуватися в клієнтському коді. В Visual C# 5.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласною конструкцією мови.

Як об’єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод `Main` – крапку входу застосунка – інкапсулюється у визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово override, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, що реалізує інтерфейси, але не підтримуюче спадкування.

На додаток до основних описаних об’єктно-орієнтованих принципів, мова Visual C# спрощує розробку компонентів програмного забезпечення завдяки декільком інноваційним конструкціям мови, у число яких входять наступні:

– Інкапсульовані підписи методів, називані делегатами, які підтримують строго-типізовані повідомлення про події.

– Властивості, що виступають у ролі методів доступу для закритих змінних-членів.

– Атрибути з декларативними метаданими про типи під час виконання.

– Вбудовані коментарі XML-документації.

– LINQ (Language-Integrated Query), що пропонує вбудовані можливості запитів у різних джерелах даних.

Якщо буде потрібно забезпечити взаємодію з іншим програмним забезпеченням Windows, таким як об'єкти COM або власні бібліотеки DLL Win32, у мові Visual C# можна використовувати процес, що називається "Interop". Процес Interop дозволяє програмам на Visual C# виконувати практично будь-які дії, які може виконувати вихідний додаток на C++. Мова Visual C# підтримує навіть покажчики й поняття "небезпечного" коду для тих випадків, коли прямий

доступ до пам'яті має вкрай важливе значення. Процес побудови Visual C# у порівнянні з C і C++ простий і є більше гнучким, чим в Java. Немає окремих файлів заголовка, а методи й типи не потрібно повідомляти в певному порядку. У вихідному файлі Visual C# може бути визначене будь-яке число класів, структур, інтерфейсів і подій.

Архітектура платформи .NET Framework

Програма мовою Visual C# виконується в середовищі .NET Framework – інтегрованому компоненті Windows, що містить віртуальну систему виконання (середовище CLR) і уніфікований набір бібліотек класів. Середовище CLR являє собою комерційну реалізацію корпорацією Майкрософт інфраструктури CLI, що є міжнародним стандартом, який лежить в основі створення середовищ виконання й розробки, у яких забезпечується тісна взаємодія між мовами й бібліотеками.

Вихідний код, написаний мовою Visual C#, компілюється в проміжну мову (IL) у відповідності зі специфікацією CLI. Код IL і ресурси, такі як растрові зображення й рядки, зберігаються на диску у файлі, що виконується, називаному складанням, з розширенням EXE або DLL у більшості випадків. Складання містить маніфест із відомостями про типи складання, версії, мови й регіональні параметри та вимоги безпеки.

При виконанні програми на Visual C# складання завантажується в середовище CLR залежно від відомостей у маніфесті. Далі, якщо вимоги безпеки дотримані, середовище CLR виконує JIT-компіляцію для перетворення коду IL в інструкції машинного коду. Середовище CLR також надає інші служби, що відносяться до автоматичного збору сміття, обробки виключень і керуванню ресурсами. Код, виконуваний середовищем CLR, іноді називають "керованим кодом" у протиставлення "некерованому коду", що компілюється в машинний код, призначений для певної системи. Далі показані відносини під час компіляції й час виконання між файлами з вихідним кодом Visual C#, бібліотеками класів .NET Framework, складаннями й середовищем CLR.

Взаємодія між мовами є ключовою особливістю .NET Framework. Оскільки код IL, створюваний компілятором Visual C# відповідає специфікації CTS, код IL на основі Visual C# може взаємодіяти з кодом, створюваним версіями мов Visual Basic, Visual C++, Visual J# платформи .NET Framework і ще більш ніж 20 CTS-сумісних мов. В одному складанні може бути кілька модулів, написаних на різних мовах платформи .NET Framework, і типи можуть посилатися один на одного, як якби вони були написані на одній мові.

Крім служб часу виконання, в.NET Framework також є велика бібліотека, що складається з більш ніж 4000 класів, організованих по просторах імен, які забезпечують різноманітні корисні функції для будь-яких дій, починаючи від введення й виведення файлів для керуванням рядками для розбивки XML, і закінчуючи елементами керування Windows Forms. У звичайному застосунку мовою Visual C# бібліотека класів .NET Framework інтенсивно використовується для "устрою" коду.

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 розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

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.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

2026 рік знаменує собою історичний поворот у нашому сприйнятті взаємозв'язку між машинами та фізичним світом. Майже десять років цифровий світ працював на зв'язку, де кожен фрагмент даних від датчика мав передаватись на віддалений хмарний сервер, перш ніж можна було прийняти рішення. Тепер цей зв'язок розірвано. Ми спостерігаємо, як інтелект переноситься з хмари на вулицю, у заводський цех та на шасі автомобілів.

Автономні системи стали фізичними об'єктами штучного інтелекту, які бачать, думають та діють у режимі реального часу. Оскільки ці системи переходять від пілотних проектів до основних міських послуг, інфраструктура, що їх підтримує, перемістилася з околиць міста на безпосередній кордон мережі.

У цьому розділі досліджується фундаментальний зсув у бік рішень для центрів обробки даних на периферії та те, як вони забезпечують низькозатримку підключення, необхідну для справжньої операційної автономії. Ми відповімо на запитання:

– Чому периферійні мережі стають кращим середовищем для критично важливих операцій?

– Як інфраструктура, визначена штучним інтелектом, замінює традиційні моделі програмного забезпечення?

– Яку роль відіграє рідинне охолодження в автономних робочих навантаженнях з високою щільністю?

Платформи, визначені штучним інтелектом, та фізичний штучний інтелект

Перехід від програмно-визначених платформ до платформ, визначених штучним інтелектом, є найважливішою зміною у 2026 році. Машини тепер

розглядаються як фізичний штучний інтелект, де платформа бачить світ, розуміє його та безпечно діє в реальних середовищах. Ці платформи спираються на спеціалізовані чіпи штучного інтелекту, які забезпечують до 40 разів вищу продуктивність, ніж попередні покоління, що робить локальний висновок складних моделей базовою вимогою. Це дозволяє автономній системі виявляти перешкоду та ініціювати реагування за мілісекунди, завдання, яке було б неможливим, якби дані мали передаватись до центральної хмари.

Індустріалізація будівництва

Розробка інфраструктури для цих систем перетворилася на гонку з часом. Стандартизоване проектування та модульне будівництво перетворили будівництво центрів обробки даних на заводський процес. Використовуючи позамайданчикове виробництво та цифрові інструменти, такі як інформаційне моделювання будівель (BIM), постачальники можуть розгортати периферійні рішення для центрів обробки даних за тижні, а не за місяці. Такий індустріалізований підхід є важливим для зниження ризиків будівництва та забезпечення того, щоб інфраструктура йшла в ногу зі швидким розгортанням автономних парків обладнання.

Рідинне охолодження високої щільності

Автономні системи залежать від графічних процесорів та прискорювачів штучного інтелекту, які генерують величезне тепло. У 2026 році проривні рішення для охолодження, такі як пряме охолодження на кристалі та занурювальне охолодження, стають стандартом, оскільки, за прогнозами, потреби в робочих навантаженнях штучного інтелекту подвояться зі 103 ГВт до 200 ГВт до 2030 року (JLL 2026 Global Data Centre Outlook). Ці технології не тільки керують тепловими потребами штучного інтелекту, але й можуть знизити споживання енергії до 30%, що робить високопродуктивні обчислення життєздатними в невеликих закритих периферійних середовищах.

Самовідновлення та агентні мережі

Управління мережею перейшло від реактивного усунення несправностей

до автономного вирішення. Агентський штучний інтелект тепер контролює з’єднання з низькою затримкою в режимі реального часу, перенаправляючи трафік та виправляючи проблеми без втручання людини. Ці системи самовідновлення гарантують, що дрон доставки ніколи не втрачає з’єднання, навіть у густонаселених міських каньйонах. На той час, коли людина помітить затримку, мережа вже перекалібрується для підтримки протоколів безпеки.

Синергія гібридної периферійної хмари

Дебати, які колись точилися навколо периферійних технологій та хмарних технологій, тепер змістилися на те, як вони працюють разом. У 2026 році хмара оброблятиме масштабне навчання базових моделей, тоді як периферійні технології виконуватимуть локалізовані, критично важливі за часом дії. Понад 40% провідних підприємств до 2028 року впровадять гібридні обчислювальні архітектури у критичні бізнес-процеси (Gartner). Цей взаємозв'язок гарантує, що автономні системи будуть як глобально інформованими, так і локально адаптованими.

Розширене підключення 5G

Широка доступність 5G Advanced забезпечує необхідну основу для периферійних обчислень. Він підтримує понад мільйон підключених пристроїв на квадратний кілометр, що дозволяє інфраструктурі розумного міста одночасно керувати тисячами автономних об'єктів. Таке підключення гарантує миттєву передачу даних по мережі, забезпечуючи наднадійний зв'язок з низькою затримкою (URLLC), необхідний для високошвидкісних протоколів безпеки.

Модульне та мікро-периферійне розгортання

Щоб досягти найменшої можливої затримки, центри обробки даних стають меншими та наближаються до місць подій. Мікростійки на периферії розгортаються в супутникових точках, базових станціях і навіть у промислових вузлах. Ці компактні пристрої є важливими для середовищ, де простір обмежений, наприклад, усередині роздрібного складу для відстеження запасів у режимі реального часу або біля основи інтелектуального світлофора.

Суверенітет даних та локалізована обробка

Нормативні вимоги тепер вимагають, щоб конфіденційні дані, такі як біометрична інформація зі штучного інтелекту в салоні, оброблялися локально. Центри обробки даних на периферії дозволяють компаніям дотримуватися цих законів про локальність даних, зберігаючи при цьому високу продуктивність. Це утримує дані користувачів у певних географічних межах, зменшуючи ризик масового витоку централізованих даних та забезпечуючи дотримання національних законів про захист.

Стала, зелена периферійна інфраструктура

Сталий розвиток зараз є операційним показником, а не другорядною метою. Сучасні периферійні об'єкти використовують відновлювану енергію та високоефективне обладнання для мінімізації свого вуглецевого сліду. Системи рідинного охолодження можуть досягати коефіцієнтів ефективності використання енергії (PUE) до 1,05 порівняно з 1,4 для об'єктів з повітряним охолодженням. Це має вирішальне значення, оскільки загальний попит на енергію від штучного інтелекту та автономних пристроїв продовжує зростати.

Цифрові двійники та віртуальне навчання

Автономні системи тепер навчаються у віртуальних середовищах ще до того, як вони вийдуть на дорогу. Ці цифрові двійники потребують величезної обчислювальної потужності на периферії для моделювання фізики реального світу та неочікуваних граничних випадків, що гарантує, що фізичний ШІ народжується з досвідом. Ці симуляції оновлюються в режимі реального часу на основі даних з польових даних, створюючи безперервний цикл навчання, який підвищує безпеку кожного підрозділу в автопарку.

Злиття штучного інтелекту, 5G та периферійних обчислень створило нову нервову систему для суспільства. Як ми бачили, головні тенденції 2026 року вказують на більш децентралізований, інтелектуальний та адаптивний світ. Від самовідновлювальних мереж до кластерів графічних процесорів з рідинним охолодженням, інфраструктура нарешті наздоганяє уяву. Розміщуючи

обчислювальну потужність саме там, де вона потрібна, ми дозволяємо автономним системам діяти з рівнем точності та безпеки, який був неможливим. Шлях від хмари до периферії завершено, і ера фізичного ШІ розпочалася.

3.2 Розробка структурної схеми

Розроблювальний у даній роботі програмний продукт Кастомізатор ЦОД Manager – це програмний продукт, що автоматизує керування різнорідної IT інфраструктурою центра обробки даних з метою кастомізації під конкретні вимоги клієнта. У цьому розділі я коротко опишу, для чого цей продукт призначений і які завдання вирішує.

Кастомізатор ЦОД – це серверна платформа, що базується на лінійках Rack і Blade.

Особливість серверної платформи програмний продукт Кастомізатор ЦОД у тому, що в конфігурації є одна блейд-кошик і деякий набір серверів. При цьому немає градації рішень і є внутрішній комутатор – модель 62 серії, Fabric Interconnect.

Традиційний підхід побудови інфраструктури полягає в тому, що в кожну блейд-кошик містяться комутатори типу LAN, SAN або Management. З такого підходу ми одержуємо безліч крапок керування й складне кабелювання. Далі все це виноситься в комутатор Top of Rack і в результаті виходить зайве накопичення. Програмний продукт Кастомізатор ЦОД Manager дозволяє вирішити проблему настільки складної організації інфраструктури.

Серверне рішення програмний продукт Кастомізатор ЦОД відрізняється від традиційної схеми побудови інфраструктури тим, що на Fabric Interconnected будується єдиний внутрішній комутатор, що має винесені порти в кожну блейд-корзину. Завдяки цьому ми зменшуємо кількість сполучних ліній між центральними комутаторами й самими лезами. Ми одержуємо більшу масштабованість і легко нарощуємо ресурси.

Пари інтерконнектів може підтримувати до 20 шасі лез. На таких інтерконнектах відноситься System Manager, що саме й взаємодіє із програмний продукт Кастомізатор ЦОД Manager, завдяки чому ми можемо управляти пулами серверів і нав'язувати на них політики.

Керування фізичною інфраструктурою й віртуальними середовищами

У продукті програмний продукт Кастомізатор ЦОД Manager усе зводиться до єдиної крапки керування. Він дозволяє нам з єдиної крапки управляти всіма чотирма рівнями: віртуалізація, обчислення, системи зберігання даних, мережа.

Структура програмного продукту Кастомізатор ЦОД Manager

Фізичне середовище:

– Сервера: Cisco, HP, IBM, Dell. Від них потрібне наявність менеджменту керування сервером.

– Системи зберігання даних: EMC, VNX, VNX2, NetApp, вся лінійка FAS.

– Мережна частина: комутатори Cisco Nexus v1000, залізні Nexus, ASA і Brocade.

Віртуальне середовище: підтримується Hyper-V, VMware. Також є інтеграція з публічними хмарами.

Ролі керування:

– Кінцевий користувач, того хто користується панеллю самообслуговування.

– IT адміністратор.

– Оператор, що займається моніторингом і базовими операціями з інфраструктурою

Система є модульною, отже є можливість інтеграції у віртуальну інфраструктуру без вимоги додаткових обчислювальних потужностей.

За допомогою програмного продукту Кастомізатор ЦОД Manager ми одержуємо глобальний моніторинг всіх чотирьох складових:

– Віртуалізація.

– Сервера.

– Мережа.

– Системи зберігання.

Рисунок 3.1 – Структурна схема системи

Все це здійснюється через користувальницький портал. Користувальницький портал надає деякий набір шаблонів, які вже зконфігуровані, і можливість відстеження виконання запитів. Через користувальницький портал можна сформувати запит на створення віртуальної віртуальної машини. Запити можуть вимагати попереднього твердження, а можуть виконуватися повністю автоматично. Їсти можливість створити віртуальну машину максимально швидко, не чекаючи дій з боку технічного фахівця. Необхідність твердження запиту залежить від пула, у якому запитувана машина буде розташована. Є обчислювальні пули, які контролюються адміністратором, а є пули, з якими користувач взаємодіє прямо, і повідомлення приходить адміністраторові тільки по факті виконання.

Програмний продукт Кастомізатор ЦОД Manager це продукт, що управляє всіма чотирма напрямками адміністрування інфраструктури й зводить їх у єдину крапку. При цьому відсутня необхідність перемикання між різними системами керування й використання консолі. програмний продукт Кастомізатор ЦОД Manager є готовим коробковим продуктом, його налаштування займає порядку години або двох залежно від глибини налаштування. Базова функціональність стає доступна буквально за 1 годину розгортання.

При кастомизації ЦОД притримуйтеся ряду порад:

– Впроваджуйте ефективне управління повітрям, щоб мінімізувати або усунути змішування повітря між секціями холодного та гарячого повітря. Це включає конфігурацію шляхів забору повітря та виведення тепла обладнанням, розташування подачі та повернення повітря, а також загальні схеми потоку повітря в приміщенні. Не забудьте створити бар'єри та герметизувати отвори, щоб уникнути рециркуляції повітря. Подавайте холодне повітря виключно в холодні проходи та забирайте гаряче повернене повітря лише з гарячих проходів.

– Організація кабелів під підлогою та над головою важлива для мінімізації перешкод у схемі охолоджувального повітря.

– Використання регулювання швидкості вентилятора для подачі лише стільки повітря, скільки потрібно IT-обладнанню, може зменшити споживання енергії до 66%.

– Ретельно продумайте розташування підлогової плитки, щоб оптимізувати розподіл повітря та запобігти короткому замиканню.

– Управління рівномірним статичним тиском у фальшпідлозі шляхом ретельного розміщення кондиціонерів забезпечує рівномірний розподіл повітря до IT-обладнання.

– Створіть конструкцію з низьким перепадом тиску, щоб мінімізувати споживання енергії вентиляторами, зробивши повітроводи якомога більшими та коротшими, а також використовуючи велику фальшполу.

– Завжди, коли це можливо, використовуйте певний економайзер, щоб заощадити гроші, енергію та захистити довкілля.

3.3 Розробка функціональної схеми

Відзначимо основні переваги програмний продукт Кастомізатор ЦОД, завдяки яким вибір припав саме на дану систему:

– Програмний продукт Кастомізатор ЦОД є уніфікованою системою, а не розрізненою групою серверів зі спробою єдиного керування;

– наявність уніфікованої фабрики, що містить вбудовується управление, що, використання універсального транспорту, відсутність рівня комутації в шасі й на рівні гіпервізора – Virtual Interface Card, простої каблювання;

– єдина конвергентна фабрика на всю систему (одноразове керування до 320 серверами) на відміну від конкурентів – своя фабрика на кожне шасі (до 16 серверів);

– одна крапка керування, швидкість конфігурування при використанні політик і профілів, що дає мінімізацію ризиків при налаштуванні, розгортанні, тиражуванні, забезпечує високу масштабованість.

Традиційні блейд системи мають наступні недоліки:

– кожний блейд-сервер – ідеологічно, з погляду керування такий же стійковий або баштовий сервер, тільки в іншому форм-факторі;

– кожний сервер – складна система з великим числом компонентів і більшим числом потужних засобів керування;

– кожне шасі, кожний сервер, і в більшості випадків кожний комутатор повинні налаштуватися окремо й найчастіше «вручну»;

– «зонтичні» «єдині» системи керування існують, але це в більшості випадків просто набір посилань на окремі утиліти;

– інтеграція «наверх» у цих систем можлива, але обмежується «стандартними» простими інтерфейсами.

На відміну від конкурентів програмний продукт Кастомізатор ЦОД має продуману систему керування – програмний продукт Кастомізатор ЦОД Manager. Даний продукт має наступні відмінні риси:

– не вимагає окремого сервера, СУБД, ОС і ліцензій на них – він частина програмний продукт Кастомізатор ЦОД;

– не вимагає інсталяції й налаштування – просто включите систему й задайте IP-адресу;

– володіє убудованої відказостійкістю;

– має ефективні й прості засоби резервного копіювання конфігурації;

– обновляється декількома «кліками»;

– делегує весь свій функціонал «наверх» через відкритий XML API;

– коштує $0.00.

Сам програмний продукт **Кастомізатор ЦОД Manager** розташовується на обох Fabric Interconnect, і, відповідно, всі налаштування, політики й т.п. зберігаються також на них.

До однієї пари Fabric Interconnect може підключено до 20 блейд-шасси. Керування кожним із шасі здійснюється з однієї крапки – програмний продукт Кастомізатор ЦОД Manager. Це можливо завдяки тому, що Fabric Interconnect будується як розподілений комутатор за участю Fabric Extender'ов . Кожне шасі має по двох Fabric Extender, з можливістю підключення 4 або 8 10GB лінками кожного.

Починаємо із самого основного: підключення консольного проведення й завдання початкової IP-адреси. Після чого запускаємо браузер і з'єднуємося. (Знадобиться java, тому що за суттю GUI програмний продукт Кастомізатор ЦОД Manager'а реалізований на ній). На малюнках нижче представлений зовнішній вигляд програми, що з'являється на екрані після уведення логіну й пароля й входу в програмний продукт Кастомізатор ЦОД Manager.

Функціональний блок «LAN»

Роботу з налаштування починаємо на вкладці «LAN». Треба відзначити, що з налаштування даної вкладки, а також вкладки «SAN» починається налаштування всієї системи. Особливо актуально це у випадку використання бездискових серверів – саме наш випадок.

Тут налаштовується абсолютно все, що має відношення до мережі: які VLAN дозволені на тому або іншому Fabric Interconnect, якими портами з'єднані Fabric Interconnect між собою, які порти в якому режимі працюють і багато чого іншого.

Також на цій вкладці налаштовуються шаблони для віртуальних мережних інтерфейсів: vNIC, які надалі будуть застосовуватися в сервісних політиках, описаних вище. На кожний vNIC (яких до слова на хості може бути до 512 при установці 2 плат програмний продукт Кастомізатор ЦОД VIC) прив'язується певний VLAN, або група VLAN і «спілкування» фізичного хоста по мережі йде саме через ці vNIC.

Організація такої кількості vNIC можлива за допомогою карти віртуального інтерфейсу VIC (Virtual Interface Card). У сімействі блейд серверів програмний продукт Кастомізатор ЦОД існує 2 види VIC: 1240 і 1280.

Кожна з них має можливість підтримки до 256 vNIC або vHBA. Головна відмінність у тому, що 1240 є LOM модулем, а 1280 мезанинна карта, так само в пропускній здатності.

1240 самостійно має можливість агрегації до 40 GB, з можливістю розширення до 80GB при використанні Expander Card.

1280 споконвічно вміє пропускати через себе до 80GB.

VIC дозволяють динамічно призначати пропускну здатність на кожний vNIC або vHBA, а так само підтримують hardware failover.

Нижче можна помітити кілька створених пулів IP адрес, з яких ці самі адреси будуть привласнюватися мережним інтерфейсам у певних VLAN (аналог EBIPA в HP). Аналогічна справа обстоїть і з MAC адресами, які беруться з MAC пулів.

У загальному концепція проста: створюємо VLAN(и), набудовуємо порти на Fabric Interconnect, набудовуємо vNIC, видаємо цим vNIC MAC/IP адреси з пулів і заганяємо в сервісну політику для подальшого використання.

Як ви знаєте існують валідовані дизайни з великими гравцями ринку систем зберігання даних, наприклад, такий як FlexPod. Дизайн FlexPod включає серверне й мережне встаткування – Cisco, а в ролі СЗД виступає NetApp. Навіщо запитаєте ви там ще й мережне встаткування? Справа все в тому, що у версії 1.4 і 2.0 програмний продукт Кастомізатор ЦОД Manager сам Fabric Interconnect не здійснював FC зонування. Для цього був необхідний вищестоящий комутатор (наприклад, Cisco Nexus), підключений до Fabric Interconnect через Uplink порти. У програмному продукті Кастомізатор ЦОД Manager є підтримка локальних зон, що дало можливо підключати Storage прямо до Fabric Interconnect. Але все-таки залишати FlexPod без комутатора не варто, тому що FlexPod є самостійним будівельним блоком ЦОДу.

Функціональний блок «SAN»

Тут розташовується все, що відноситься до мережі передачі даних: iSCSI і FCoE. Концепція така ж, як і в мережних адаптерів vNIC, але зі своїми відмінностями: WWN, WWPN, IQN і т.п.

Функціональний блок «Equipment»

За допомогою даного інтерфейсу можна побачити інформацію про нашу загальну обчислювальну інфраструктуру, що була розгорнута для тестування. Так, було задіяно одне шасі (Chassis 1) і чотири сервери-леза (Server1, Server2, Server3, Server4). Цю інформацію можна одержати з дерева в лівій частині (Equipment), або в графічному виді (Physical Display). Крім кількості шасі/серверів можна подивитися інформацію про допоміжні компоненти (вентилятори, блоки живлення й т.п.), а також інформацію про різний тип попереджень. Загальна кількість алертов виводиться в блоці «Fault Summary», окремо інформацію з кожного вузла можна подивитися, кликнувши по ньому ліворуч. Кольорові рамки навколо назв вузлів ліворуч допомагають довідатися, на якому вузлі є алерти. Наші Fabric Interconnect налаштовані на роботу в failover режимі (при виході з ладу одного з агрегатів – його роботу негайно візьме на себе іншої). Інформація про це видно аж унизу дерева – позначення primary і subordinate.

Рисунок 3.2 – Функціональна схема системи

Функціональний блок «Servers»

Тут перебувають різні політики, шаблони сервісних профілів і безпосередньо самі сервісні профілі. Наприклад, ми можемо створити кілька організацій, кожної з яких призначити свої політики вибору завантажувального пристрою, доступу до певних VLAN/VSAN і ін.

Глобальний зміст усього цього: попередньо налаштувавши все політики й шаблони, можна швидко набудовувати нові сервера для роботи, застосовуючи той або інший сервісний профіль (Service Profile).

Сервісний профіль, що містить у собі вищезгадані налаштування, може бути застосований до одного певного сервера, або групі серверів. Якщо в ході розгортання виникають які-небудь помилки або повідомлення – ми можемо подивитися їх у відповідній вкладці «Faults»:

Після чого користувачі цих організацій можуть розгорнути й настроїти сервер під себе з обраною операційною системою й набором передвстановленим набором програмного забезпечення. Для цього необхідно скористатися можливостями інтерфейсу програмний продукт Кастомізатор ЦОД Manager, або звернувшись до автоматизації: наприклад за допомогою продукту VMware vCloud Automation Center або Cisco Cloupia.

Функціональний блок «VM»

В VMware vCenter можна встановити плагіни для програмний продукт Кастомізатор ЦОД Manager. Так, ми підключили плагін, через який підключили віртуальний дата-центр із назвою «FlexPod_DC_1». У ньому працює Cisco Nexus 1000v (nexus-dvs), і ми маємо можливість дивитися його налаштування.

Функціональний блок «Admin»

Тут ми можемо подивитися глобальні події: помилки, попередження. Зібрати різні дані у випадку звернення до підтримки Cisco TAC і т.п. речі. Зокрема їх можна одержати на вкладках «TechSupport Files», «Core Files».

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.

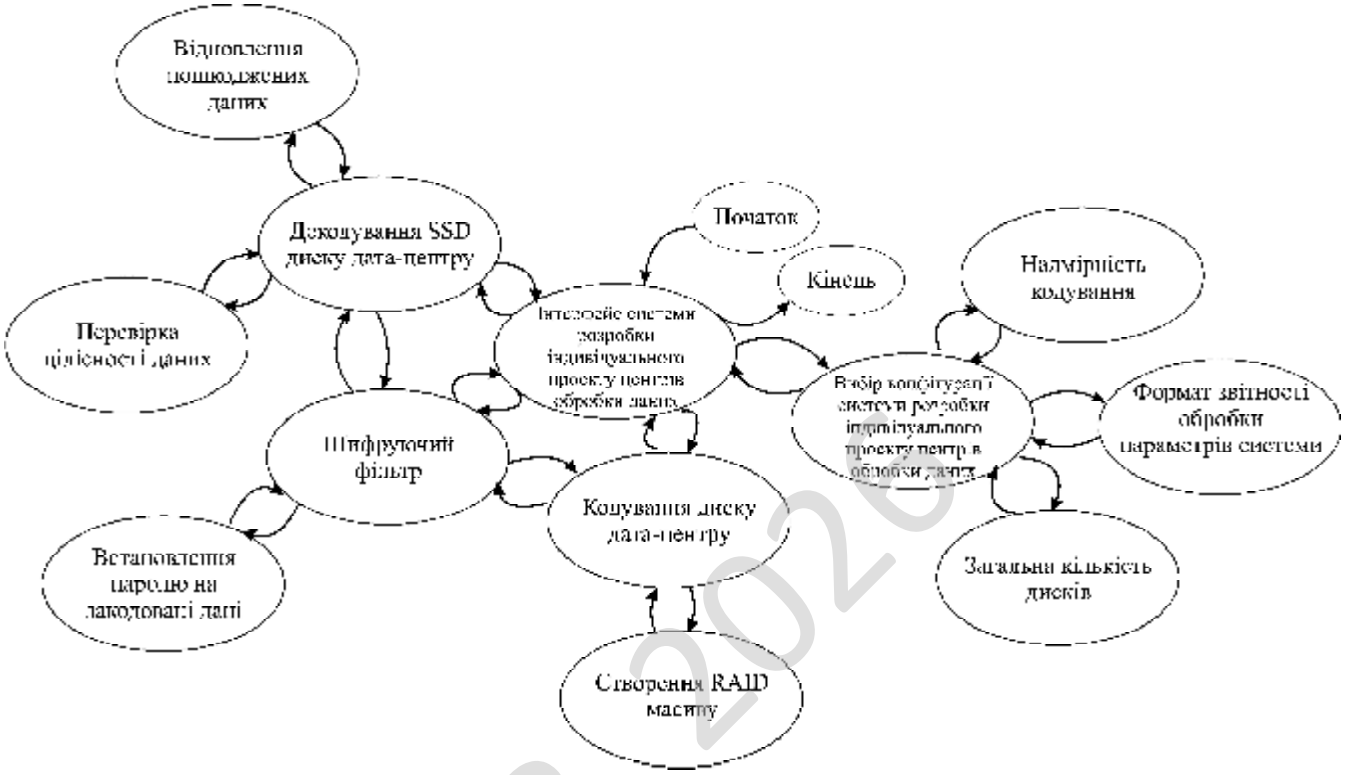

Рисунок 3.3 – Діаграма взаємодії процесів

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.

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

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

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

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

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

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

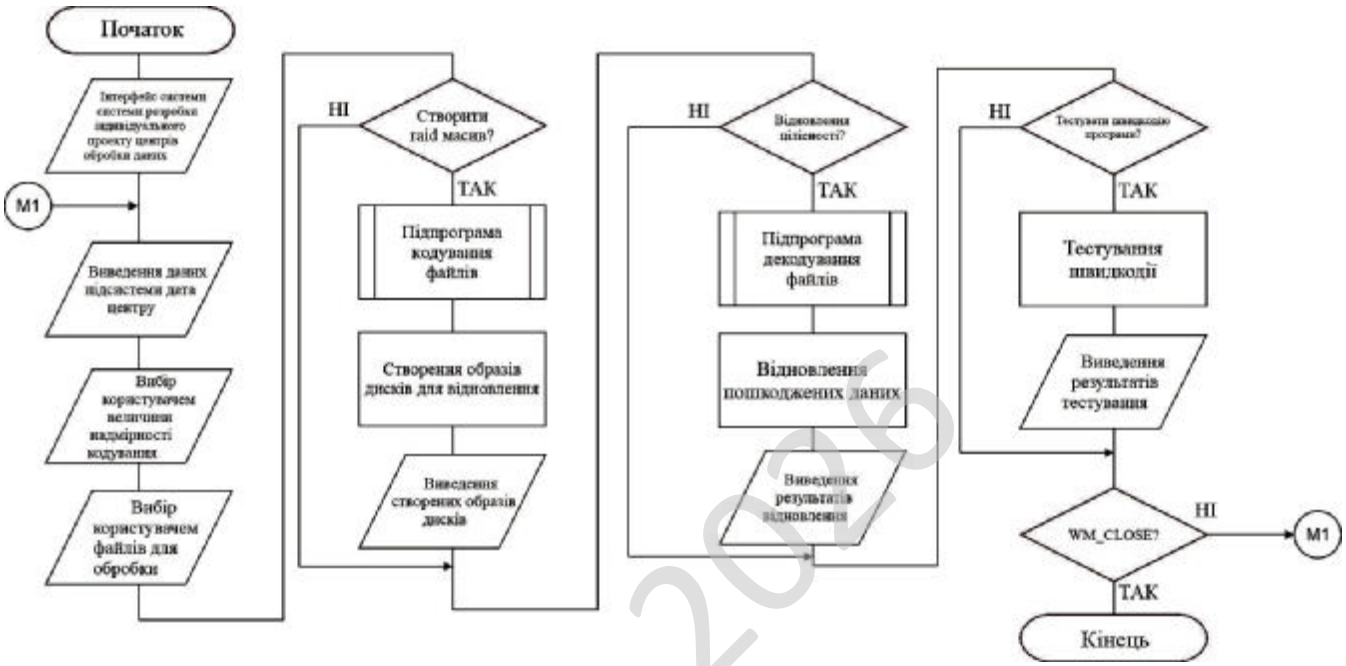

Рисунок 4.1 – Блок-схема основної програми

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.

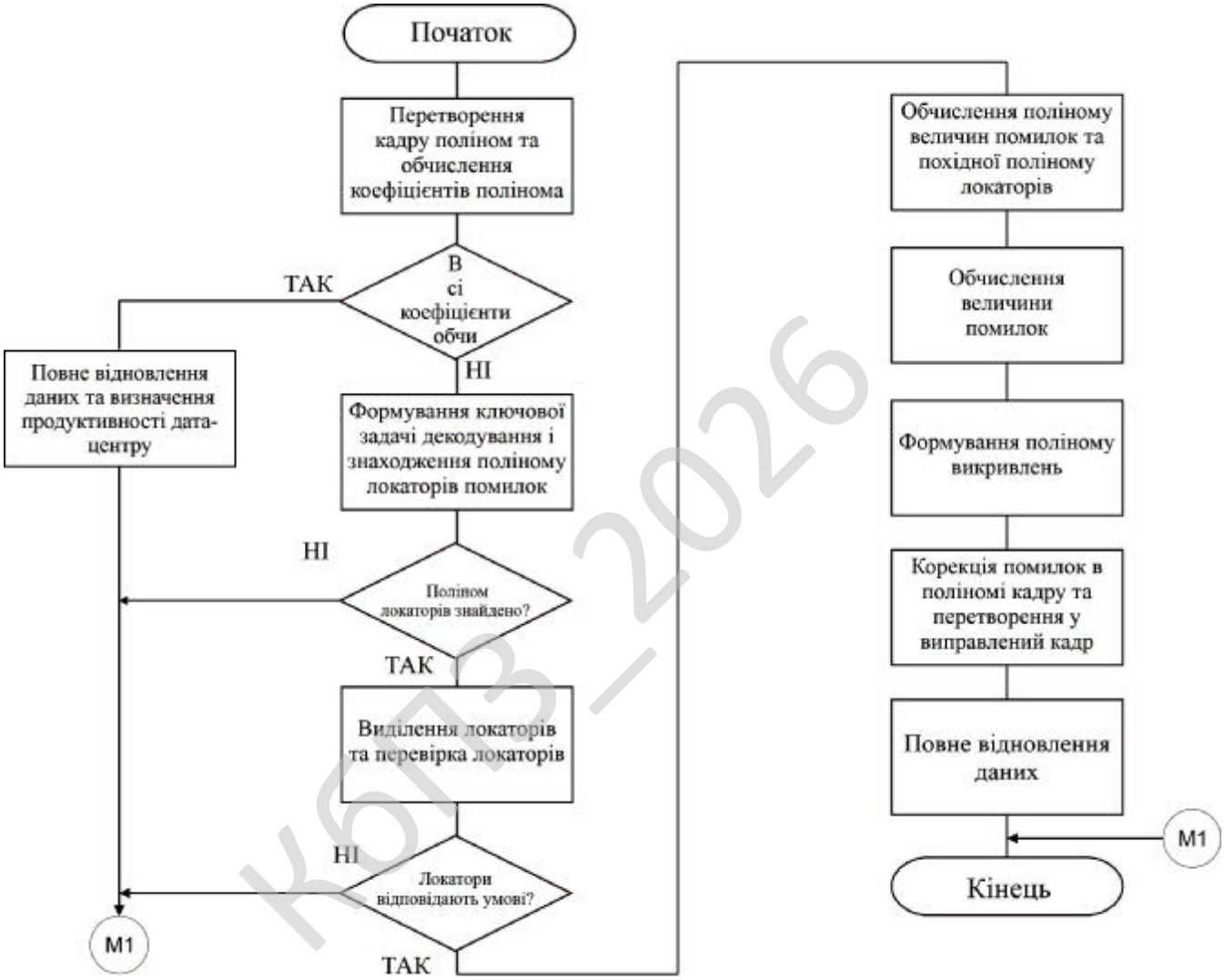

Рисунок 4.2 – Блок-схема роботи підпрограми

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

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

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

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

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

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

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

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

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

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

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

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.

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

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

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

Основні характеристики СКБД

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

7. Підвищений рівень безпеки.

Можливості СКБД

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

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

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

Основні компоненти середовища СКБД

1. Апаратне забезпечення.

2. Програмне забезпечення.

3. Дані.

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

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

Архітектура СКБД

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

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.

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.

Firebird (також Firebird SQL) – компактна, крос-платформова, вільна реляційна система керування базами даних, що реалізує більшість функцій стандарту SQL 2003. Вона може запускатись на більшості UNIX-подібних систем (в тому числі Linux та FreeBSD) та Windows.

Основні можливості

Відповідність вимогам ACID: Firebird спеціально спроектовано таким чином, щоб задовільняти вимоги «атомарності, несуперечності, ізоляції та довговічності» транзакцій «Atomicity, Consistency, Isolation and Durability».

Версійна архітектура: основна особливість Firebird – версійна архітектура, що дозволяє серверу обробляти різні версії одного запису в будь-який час таким чином, що кожна транзакція бачить свою версію даних, не заважаючи сусіднім. Таким чином, транзакції, що читають, не блокують транзакції, що пишуть і навпаки. Окрім того, це дає можливість відмовитись від логу транзакцій і таким чином зменшити ймовірність пошкодження службової інформації бази даних.

Збережені процедури: за допомогою мови PSQL (процедурна SQL) можна створювати складні збережені процедури для обробки даних на боці сервера. Таким чином можна виносити на сторону сервера значну частину бізнес-логіки програмного пакету чи формувати дані для звітів.

Події: збережені процедури та тригери можуть ґенерувати події, на які, в свою чергу, може підписатися клієнтська програма і відповідним чином їх обробляти.

Генератори: дають можливість просто реалізовувати автоінкрементні поля; оскільки вони працюють незалежно від транзакцій, то можуть використовуватись для генерації первинних ключів чи керування тривалими запитами в інших транзакціях.

Бази даних в режимі «лише читання»: спрощують поширення даних наприклад на компакт-дисках, особливо в поєднанні з вбудованою (embedded) версією сервера.

Повний контроль над транзакціями: одна клієнтська програма може одночасно виконувати декілька транзакцій, включно з різними рівнями ізоляції. Окрім того, доступні протокол двофазного підтвердження транзакцій (що забезпечує гарантовану стійкість при роботі з кількома БД), оптимістичне блокування даних (блокується не вся сторінка даних, а лише змінені записи), точки збереження транзакцій та автономні транзакції (починаючи з версії 2.5).

Резервне копіювання на лету: завдяки в тому числі і версійній архітектурі немає потреби зупиняти сервер для резервування бази даних. Процес резервного копіювання зберігає стан бази даних на момент початку резервування, не шкодячи роботі інших клієнтів. Додатково існує можливість інкрементного резервування бази даних (починаючи з версії 2.0).

Тригери: для будь-якої таблиці можна назначити декілька тригерів, що спрацюють до чи після додавання, оновлення чи вилучення даних. У тригерах використовується мова PSQL, що дозволяє задавати початкові значення даних, перевіряти їх цілісність, збуджувати події та ін. Починаючи з версії 1.5 у Firebird з'явились універсальні тригери, що дозволяють обробляти вставку, оновлення та вилучення даних в одному місці.

Зовнішні функції: можна реалізувати за допомогою будь-якої мови програмування та у вигляді бібліотек користувацьких функцій (англ. UDF – User Defined Function) під'єднаних до сервера.

Набори символів: Firebird підтримує багато міжнародних наборів символів з багатьма варіантами сортування, зокрема типовий для українських користувачів

Windows набір символів Win1251 підтримує три варіанти сортування, в тому числі win1251_ua, що дозволяє коректно сортувати отримані з бази дані з українськими символами на боці сервера. Окрім того, Firebird повністю підтримує Unicode, що дає можливість працювати з будь-якими наборами символів.

Firebird повністю підтримує стандарт SQL-92 та реалізує більшу частину стандартів SQL-99 і SQL-2003. Сюди входять вирази DML/DDL, об'єднання запитів, вирази UNION, DISTINCT, підзапити (IN, EXISTS), агрегатні функції (AVG, SUM, MIN, MAX, LIST (починаючи з версії 2.1)), вбудовані функції (ABS, CEIL, REPLACE, GEN_UUID тощо), обмеження цілісності (PRIMARY KEY, UNIQUE, FOREIGN KEY), та всі загальні типи даних SQL.

Особливості доступу та оброблення даних в Firebird

PSQL (Procedural SQL) – підмножина SQL в Firebird, за допомогою якої пишуться збережені процедури та тригери. Надає можливість програмісту обробки даних у процедурному стилі – наприклад, за допомогою циклів.

```
CREATE OR ALTER PROCEDURE SOME_PROC(
    IN_ID INTEGER)
RETURNS (
    OUT_ID INTEGER)
AS
begin
   while (IN_ID < 10) do
    begin
     OUT_ID = IN_ID;
     IN_ID = IN_ID + 1;
     suspend;
    end
end
```

DSQL (Dynamic SQL) – підмножина SQL, за допомогою якої здійснюються запити до даних. Підтримуються неіменовані параметри.

```
select some_field1 from some_table where some_field2=? and some_field3
```

Доступ до баз даних Firebird може здійснюватися через розподілену бібліотеку доступу (dll або so, залежно від платформи) – такий спосіб є стандартним; сама бібліотека є в комплекті дистрибутиву. Також є можливість доступу через java і .net провайдери.

Бібліотека доступу реалізує набір функцій для маніпуляції з даними, які можуть бути експортовані і використані в будь-якій мові програмування.

Окрім того, є достатньо багато обгорток під різні мови програмування для цієї бібліотеки, що звільняють програміста від рутинної низькорівневої роботи. Так, для Delphi популярними є бібліотеки IBX, FibPlus, UIB. Для C++ – IBPP. Підтримка вбудована і в популярні скриптові мови, такі як PHP і Python.

Обмеження Firebird

Розмір бази даних – необмежено (залежить від можливостей файлової системи). Розмір таблиці бази даних – 32 ТБ. Максимальна ширина вибірки (сумарно всі поля; не враховуючи блоби; враховується фактичний розмір рядкових даних) – 64 Кб. Індексів на таблицю – 850. Довжина об'єкта метаданих (назва таблиці, процедури тощо) – 27 символів.

Апаратно-програмні вимоги та обмеження

Firebird існує у версіях для Unix (Linux, FreeBSD, Solaris, MacOS, HP-UX) та Windows і вимоги до апаратного забезпечення залежатимуть також від типу ОС, котра обслуговує сервер. Окрім того вимоги знаходяться в прямій залежності від очікуваного завантаження сервера баз даних, обсягу оброблюваних даних та кількості одночасно працюючих користувачів і говорити про конкретні цифри непросто. Проте загалом ці вимоги доволі низькі: при незначних навантаженнях та обсягах баз даних можна очікувати пристойної роботи на сервері з центральним процесором частотою 100–200 МГц та обсягом оперативної пам'яті 96-128 МБ.

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile – Extreme Programming.

Екстремальне програмування (Extreme Programming, далі XP) це методологія розробки програмного забезпечення, найпопулярніша серед так званих гнучких методологій. Має на меті поліпшення якості програмного забезпечення та чутливість до змін у вимогах замовників. Як вид гнучких

методологій, XP радить часті "випуски" програми у коротких циклах розробки, що має на меті поліпшити продуктивність праці та покращити можливості виконання вимог замовника що змінюються. Авторами даної методології є Кент Бек, Ворд Каннінгем, Мартін Фаулер та інші.

Інші елементи екстремального програмування включають в собі: парне програмування, проведення обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того як вона дійсно необхідна, простота та ясність коду, очікування на зміну вимог замовників з плином часу та коли вимоги до продукту стають ясніші, досить часте спілкування із замовником та між самими програмістами.

Назва методології походить від ідеї застосувати корисні методи і практики розробки програмного забезпечення, піднявши їх до "екстремальних" рівнів.

Критики XP зауважують на потенційні недоліки цієї методології – нестабільні вимоги, незадокументовані компроміси конфліктів користувачів, відсутність загального документу дизайну програми.

Технологія екстремального програмування була розроблена Кентом Беком, Уардом Каннінґхемом та Роном Джеффріесом під час роботи над Chrysler Comprehensive Compensation System (C3). У 1996 Кент Бек став лідером проекту і почав вдосконалювати методи розробки, що застосовувалися в роботі над проектом. Свій метод він виклав у книзі «Extreme Programming Explained», котру було видано у жовтні 1999. Після купівлі Крайслера компанією Даймлер–Бенц проект C3 було скасовано у лютому 2000.

Хоча саме екстремальне програмування є відносно новим, багато її практик вже існували і використовувались протягом певного часу; однак, методологія підносить "найкращі практики" до екстремального рівня. Для прикладу, практика по плануванню і написанню тестів перед написанням кожної маленької частини коду було використано раніше в проекті НАСА "Меркурій". Для зменшення часу на розробку ПЗ деякі формальні документи тестування (такі як приймальне тестування) писались паралельно (або й раніше) з написанням

самого ПЗ. Незалежна група тестування НАСА може писати процедури тестування базуючись на формальних вимогах до продукту до того як програмне забезпечення розроблене та інтегроване в систему. В XP ця концепція піднесена до "екстремального рівня" завдяки написанню автоматичних тестів які перевіряють поведінку навіть малих частинок коду, а не тільки значних функціональних частин ПЗ.

Посібник Extreme Programming Explained: Embrace Change описує Екстремальне Програмування, як:

– Спроба примирити гуманність і продуктивність.

– Механізм для соціальної зміни.

– Шлях до удосконалення.

– Стиль розвитку.

Дисципліна розробки програмного забезпечення.

Головною метою Екстремального Програмування є скорочення вартості неочікуваних змін. У традиційних методах розробки (на кшталт SSADM) вимоги до розвитку системи визначаються на початку роботи над проектом, і часто виправляються пізніше. Це означає, що вартість проекту через зміни буде більшою за заплановану (традиційна особливість для програмного забезпечення, що проектується).

XP використовується для скорочення вартості змін, завдяки представленню простих значень, принципів і методів. При використанні екстремального програмування, проект повинен стати гнучкішим щодо змін.

Extreme Programming Explained описує екстремальне програмування як дисципліну розробки програмного забезпечення яка змушує людей створювати високоякісне ПЗ якомога швидше. XP намагається зменшити ціну зміни вимог до ПЗ завдяки малим циклам розробки, а не одним довгим циклом. Екстремальне програмування сприймає зміни до вимог як звичайні, неминучі та бажані аспекти розробки ПЗ, і ці зміни мають бути очікуваним. Основна ідея полягає в тому що неможливо розробити самодостатній пакет вимог до ПЗ, зміни в вимогах – неминучі. Екстремальне програмування також вводить набір практик та

принципів на основі методології гнучкої розробки програмного забезпечення. Екстремальне програмування описує чотири базові активності що виконуються при розробці програмного забезпечення: написання коду, тестування, слухання та дизайн.

Написання коду. Прихильники XP заявляють що єдиним дійсно важливим результатом розробки ПЗ є код: без готового коду нема продукту.

Тестування. Методологія екстремального програмування заявляє, що якщо дрібне тестування може перевірити незначну частину функціональності, то багато дрібних тестів можуть перевірити набагато більше частинок і продукт в цілому.

Основні прийоми XP. Дванадцять основних прийомів екстремального програмування (за першим виданням книги Extreme programming explained) можуть бути об'єднані в чотири групи:

1. Короткий цикл зворотного зв'язку (Fine scale feedback).

1.1. Розробка через тестування (Test driven development).

1.2 Гра в планування (Planning game).

1.3. Замовник завжди поруч (Whole team, Onsite customer).

1.4 Парне програмування (Pair programming).

2. Безперервний, а не пакетний процес.

2.1 Безперервна інтеграція (Continuous Integration).

2.2 Рефакторинг (Design Improvement, Refactor).

2.3 Часті невеликі релізи (Small Releases).

3. Розуміння, що поділяється всіма учасниками.

3.1 Простота (Simple design).

3.2 Метафора системи (System metaphor).

3.3 Колективне володіння кодом (Collective code ownership) або обраними шаблонами проектування (Collective patterns ownership).

3.4 Стандарт кодування (Coding standard or Coding conventions).

4. Соціальна захищеність програміста (Programmer welfare), а саме 40 годинний робочий тиждень (Sustainable pace, Forty hour week).

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму DES. DES є класичною мережею Фейштеля із двома гілками. Дані шифруються 64-бітними блоками, використовуючи 56-бітний ключ. Алгоритм перетворить за кілька раундів 64-бітний вхід в 64-бітний вихід. Довжина ключа дорівнює 56 бітам. Процес шифрування складається із чотирьох етапів. На першому з них виконується початкова перестановка (IP) 64-бітного вихідного тексту (забілювання), під час якої біти переставляються у відповідності зі стандартною таблицею. Наступний етап складається з 16 раундів однієї й тої ж функції, що використовує операції зрушення й підстановки. На третьому етапі ліва й права половини виходу останньої (16-й) ітерації міняються місцями. Нарешті, на четвертому етапі виконується перестановка IP^{-1} результату, отриманого на третьому етапі. Перестановка IP^{-1} інверсна початковій перестановці. Праворуч на рисунку показаний спосіб, яким використовується 56-бітний ключ. Спочатку ключ подається на вхід функції перестановки. Потім для кожного з 16 раундів підключ K_i є комбінацією лівого циклічного зрушення й перестановки. Функція перестановки та сама для кожного раунду, але підключи K_i для кожного раунду виходять різні внаслідок повторюваного зрушення біт ключа.

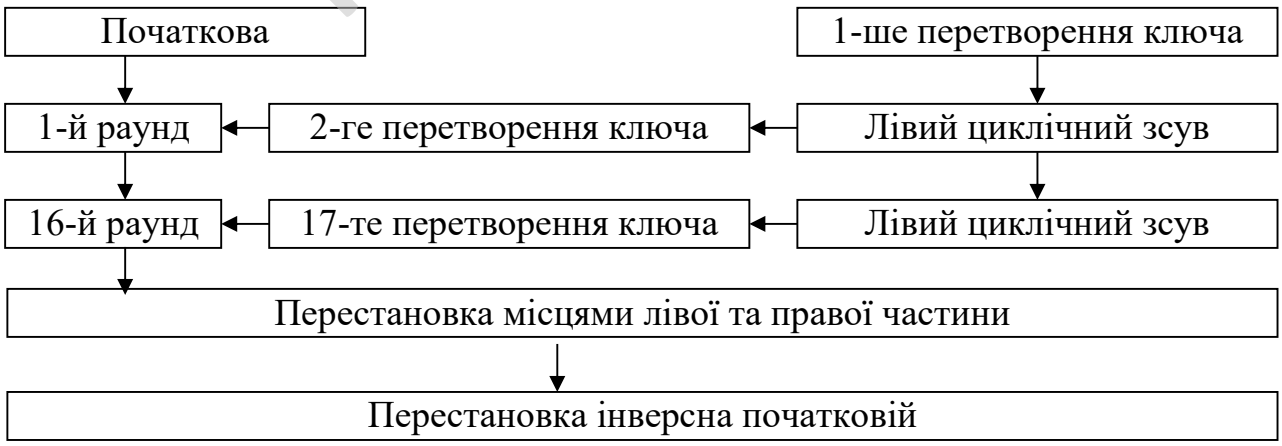

Рисунок 4.3 – Загальна схема DES

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблена система розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

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

– Верхнього меню: Файл; Інструменти; Параметри; Довідка.

– Функціональних кнопок ПЗ.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Розділу обрання групи файлів.

– Розділу виведення результату роботи системи.

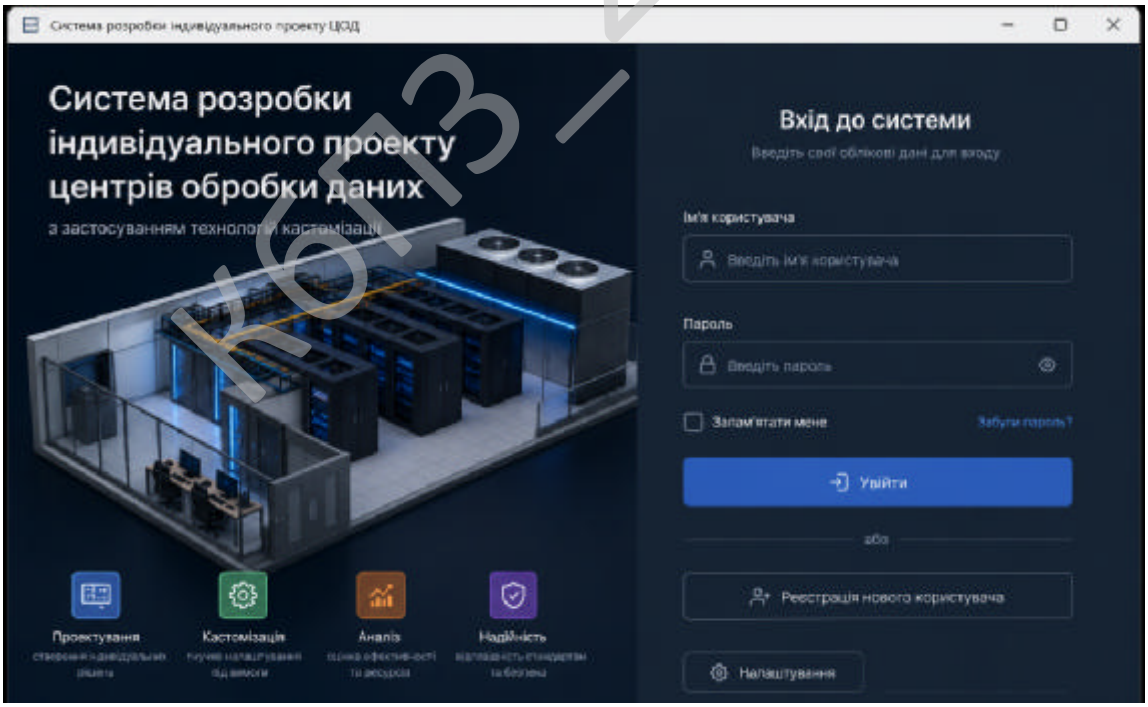

Рисунок 5.1 – Головне вікно ПЗ

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

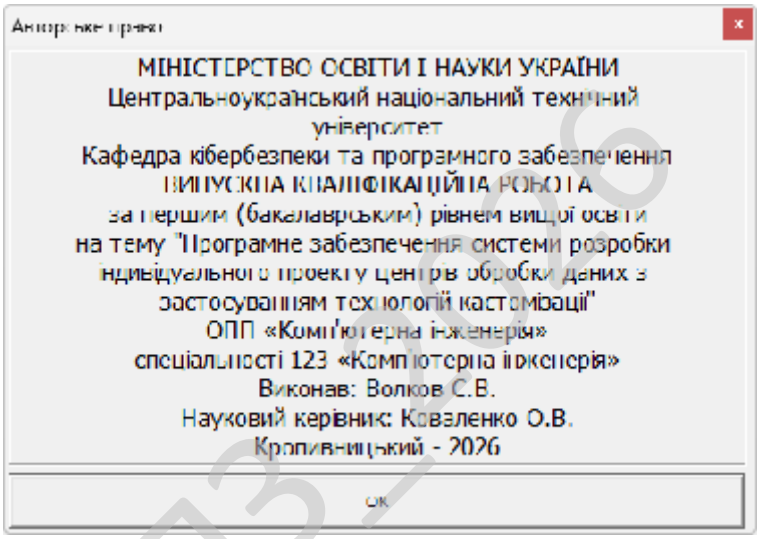

Рисунок 5.2 – Авторське право

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

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є

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

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– 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 IT рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене IT рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

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

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються

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

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

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

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

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

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

Проводилось тестування 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.

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

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

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

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

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

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

Проводилось тестування чорної скриньки.

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

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

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

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

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

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

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

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

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ITc;

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору OT.

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

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій.

– Помилок інтерфейсу.

– Помилок у зовнішніх структурах даних або в доступі до зовнішньої бази даних.

– Помилок характеристик (необхідна ємність пам'яті і т.д.).

– Помилок ініціалізації та завершення.

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.

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.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

– Досліджена система розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

– На основі отриманих результатів досліджень створена програмна реалізація системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації.

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 Visual C#. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

системи розробки індивідуального проекту центрів обробки даних з застосуванням технологій кастомізації. 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 DES.

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.

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.

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.

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

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

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

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

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

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

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

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

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

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

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

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

29. Smirnov O. Kuznetsov A., 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*

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

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

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

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

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

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

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

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

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

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

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

42. Смірнов О.А., 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

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.

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.