

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 аналізу трафіку віртуальної
інфраструктури”**

Виконав здобувач вищої освіти
IV курсу, групи КІ-23-МБ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Грицюк В.О.
«___» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Петренюк В.І.
«___» _____ 2026 р.
Рецензент _____

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

Грицюку Вадиму Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи аналізу трафіку
віртуальної інфраструктури

2. Керівник роботи Петренюк Володимир Ілліч, кандидат фізико-математичних
наук, доцент

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

затверджені наказом вищого навчального закладу № 126-02 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 аналізу трафіку віртуальної інфраструктури

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

Петренюк В.І.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Грицюк В.О.
(прізвище та ініціали)

АНОТАЦІЯ

Грицюк В.О. Програмне забезпечення системи аналізу трафіку віртуальної інфраструктури. 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 аналізу трафіку віртуальної інфраструктури.

Метою розробки є програмне забезпечення системи аналізу трафіку віртуальної інфраструктури.

Результат роботи – програмна реалізація системи аналізу трафіку віртуальної інфраструктури.

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

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

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

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, аналіз трафіку, віртуальна інфраструктура

ABSTRACT

Hrytsiuk V.O. Software for the virtual infrastructure traffic analysis system. 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 virtual infrastructure traffic analysis system.

The purpose of the development is software for the virtual infrastructure traffic analysis system.

The result of the work is the software implementation of the virtual infrastructure traffic analysis system.

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

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

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

The program was developed in the Python environment.

Keywords: computer engineering, traffic analysis, virtual infrastructure

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	28
2.3 Розгорнута постановка завдання	33
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	35
3.1 Опис функціонування системи	35
3.2 Розробка структурної схеми.....	37
3.3 Розробка функціональної схеми	45
3.4 Розробка діаграми процесів.....	49
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	52
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	52
4.2 Захист розробленого програмного забезпечення.....	69
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	71
6 ОСНОВНІ ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76

					ВКРБ-123.26.0029.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи аналізу трафіку віртуальної інфраструктури	Літ.	Аркуш	Аркушів
Розроб.	Грицюк В.О.					Б	1	83
Перев.	Петренко В.І.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23-МБ		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВМ	—	віртуальні машини
ІТ	—	інформаційні технології
ПЗ	—	програмне забезпечення
ТСKM	—	телекомунікаційні системи і комп'ютерні мережі
DFW	—	Distributed Firewall
ESX/ESXi	—	вузли/гіпервізори
QoS	—	механізми контролю й керування якістю
vMA	—	засіб збору статистики

К6ПЗ_2026

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Моніторинг віртуального середовища надає можливості моніторингу, установки значень порогів і складання звітів про інфраструктуру для забезпечення моніторингу віртуальних машин, вузлів і серверів. Для віртуальних серверів і віртуальних машин виявлення працює інакше, чому для фізичних пристроїв. Створіть профіль виявлення, що дозволяє виявляти служби керування або інші окремі гіпервізори, коли ці гіпервізори працюють у некерованому середовищі. Виявлення віртуальних машин по їхніх службах керування або гіпервізорам забезпечує продовження моніторингу й внесення даних у журнал і є особливо корисним у середовищах з високою доступністю, у яких віртуальні машини працюють по відказостійким схемам або під керуванням менеджерів робочого навантаження.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи аналізу трафіку віртуальної інфраструктури.

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

- Огляд існуючих систем аналізу трафіку віртуальної інфраструктури.
- Дослідження системи аналізу трафіку віртуальної інфраструктури.
- Програмна реалізація системи аналізу трафіку віртуальної інфраструктури.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі аналізу трафіку віртуальної інфраструктури.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи аналізу трафіку віртуальної інфраструктури, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Обов'язкові умови для сканування виявлення

Необхідні умови

Перед запуском сканування виявлення у віртуальному середовищі переконаєтеся в наявності наступного:

VMware. Використовуються дійсні облікові дані VMware. Так само потрібен сервіс VMware.

Hyper-V. Пристрої Hyper-V виявляються, якщо використовуються дійсні облікові дані Windows. Для минаючих об'єктів керування WMI необхідні групи й користувачі. Правила брандмауерів застосунку ОС вузла повинні відповідати стандартним правилам брандмауера Hyper-V.

Виявлення віртуальних середовищ від основних до другорядних

При виявленні віртуальних ресурсів у специфікацію сканування рекомендується включити IP-адреси VCenter і гіпервізорів, а не окремі VM. Whatsup Gold виявляє окремі віртуальні машини за допомогою елементів керування (таких як VCenter, вузли Hyper-V і окремі гіпервізори) у рамках процесу виявлення. При включенні віртуальних середовищ у сканування виявлення виконуйте наступні рекомендації:

- Спочатку необхідно виконати виявлення служб керування (наприклад, VMware VCenter).
- У некерованих середовищах (без VCenter і одного або більш гіпервізорів) можна виявляти окремі гіпервізори.
- Після завершення сканування виявлення застосуєте моніторинг до всіх вузлів, що ставляться до віртуальних середовищ (для кожного VCenter, гіпервізора або віртуальної машини).

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Щоб забезпечити, що зв'язок між VMware VCenter, кластером, дата-центром, гіпервізором і віртуальною машиною розпізнана Whatsup Gold і пізніше відображена на карті мережі, після виявлення необхідно застосувати/оновити моніторинг (Пуск/відновлення моніторингу) як для елементів керування, гіпервізорів, так і для зв'язаних віртуальних машин.

1.2 Область застосування

Запуск виявлення й застосування моніторингу.

Виявлення в середовищах VMware

При запуску сканування виявлення найкраще настроїти сканування на збір інформації про гіпервізорах і віртуальних машинах VMware за допомогою служби керування (таких як VMware VCenter). У некерованих віртуальних середовищах з окремими гіпервізорами (наприклад, тільки з VMware ESXi) можна виконувати виявлення окремих гіпервізорів.

Якщо керування вузлами виконується за допомогою VCenter.

Якщо вузли/гіпервізори (ESX/ESXi) працюють як окремі (некеровані).

Для того, щоб виявити VM або гіпервізор необхідно вказати облікові дані WMI (Hyper-V) або VMware під час сканування. Інакше категорія пристрою або «Роль», яку «побачить» Whatsup Gold, буде обмежена набором простих об'єктів SNMP. Якщо кінцеві точки віртуальних машин у мережі відрізняються від фізичних пристроїв, використовуваних у якості гнучких ресурсів, і звичайно надаваних по запиту, Whatsup Gold виконує періодичне Завдання по синхронізації віртуального середовища для підтримки актуальності інформації пристрою. Кількість VM у карті пристрою є актуальними на момент останнього періодичного сканування.

Карта/організація/монітор (карта Моя мережа)

Віртуальні сервери й віртуальні машини можна розглядати й спостерігати, як фізичні пристрої, але в них окремі ролі, додаткові можливості моніторингу,

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

додаткові вимоги запису в журнал і окремі фільтри (накладення віртуальних машин), що дозволяє сфокусуватися тільки на віртуальних пристроях при перегляді їх на карті Моя мережа:

- Показати пристрою Моніторинг віртуального середовища. Виберіть елемент Накладення віртуальних машин.
- Додавання й включення окремих моніторів. Виберіть пристрій, виберіть властивості пристрою, потім Призначити й включити окремі монітори.
- Запис подій у журнал. Запис подій віртуальної платформи в журнал є вбудованою функцією, що настраюється (Монітор журналу подій Hyper-V / Запис у журнал подій VMware).
- Установка елементів керування оповіщеннями. Пороги оповіщення (Hyper-V / VMware), Оповіщення, Розклад відключення й Політики оповіщення.

Аналіз (панель, журнал і звіт)

Аналіз журналу, створення звітів і даних панелі для оцінки продуктивності, якості обслуговування й впливу.

- Панель моніторингу віртуального середовища. Вибір вузла Hyper-V або VMware і перегляд метрик і відомостей про систему для зв'язаних VM.
- Головна панель. Додавання вікон, що настраюються, і налаштування й перетаскування звітів продуктивності й обліку Моніторинг віртуального середовища з палітри звітів.
- Журнал подій VMware. Керування журналами VCenter. Настроюється у властивостях пристрою.
- Журнал подій Hyper-V. Події віртуальної машини Hyper-V настраюються в налаштуваннях монітора.

Захват/демонстрація/експорт

Згенеровані дані звіту Моніторинг віртуального середовища можна візуалізувати, надати до них загальний доступ і використовувати повторно, скориставшись експортом даних або параметрами панелі.

- PDF. Збереження вікна «таблиця» і «діаграма» як PDF.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

– Демонстрація. Демонстрація звітів Моніторинг віртуальних машин і інфраструктури для інших користувачів при одиночному або груповий перегляді.

– Експорт. Експорт даних Моніторинг віртуальних машин і інфраструктури як PDF, XML або Excel використовувати за межами Whatsup Gold

– Електронна пошта. Звіт по електронній пошті, діаграми й дані журналів.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи аналізу трафіку віртуальної інфраструктури, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ_2026

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

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

Навіть невеликий прорахунок у зміні параметрів однієї ВМ може привести до зниження продуктивності й навіть відмові вузла ESX.

От чому важливо постійно тримати під контролем продуктивність віртуального середовища.

Нам необхідно аналізувати всі параметри продуктивності ВМ і ESX-хостів і одержувати повну картину, необхідну для оптимізації й підстроювання параметрів віртуального середовища.

Допомогти нам у цьому можуть системи керування й моніторингу.

Ми постаралися класифікувати засоби моніторингу й керування віртуальними середовищами й вивести загальні рекомендації (рецепти) для керування віртуальним середовищем. Нижче ми представили «рецепти» для наступних випадків:

- Застосунок для моніторингу 1-5 вузлів ESX.
- Застосунок для моніторингу 6-19 вузлів ESX.
- Застосунок для моніторингу 20 і більш вузлів ESX.
- Безкоштовні засоби моніторингу.
- Застосунок для моніторингу розподіленої мережі вузлів.
- Засоби моніторингу із пропозиціями по оптимізації.
- Засоби швидкого реагування.
- Засоби, що підходять як для віртуального середовища, так і для фізичного середовища.
- Керування продуктивністю й моніторинг застосунків.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Рецепти ці умовні, а критерії вибору однаково визначати Вам.

Після порівняння засобів моніторингу, які, на нашу думку, підходять для рецепта, ми даємо нашу рекомендацію.

Впливати їй чи ні, знову ж, вирішувати Вам.

Застосунок для моніторингу 1-5 вузлів ESX

При виборі ми враховували такі параметри, як: ціну, можливості, легкість у керуванні, а також підтримку в Україні.

При виборі застосунків для 1-5 вузлів ми виходили з того, що інфраструктура невелика, кількість віртуальних машин може досягати 50.

Потужні системи моніторингу й керування застосовувати не слід, так як більшість функцій потужних систем не будуть використовуватися.

Кількість об'єктів моніторингу розраховується, як якщо на кожному вузлі ESX буде до 10 VM.

Основними критеріями вибору були наступні: низька ціна (або безкоштовний продукт), простий і зрозумілий інтерфейс, мінімальні ресурси, виділювані для системи моніторингу.

Аналіз представлений наступною структурою:

Назва продукту – Переваги – Недоліки – Що можна побачити – Процес моніторингу

Management Assistant – Веде автоматизований моніторинг мережі. – Споконвічно працює автоматизоване тільки з 1 хостом, для того щоб велися логи всіх хостів, прийде писати численні скрипти. – Оскільки vMA є засобом збору статистики, на виході ми отримуємо .csv файл із численними даними про сервер ESX і VM на ньому. – vMA погано придатний для аналізу, оскільки аналіз повинен здійснюватися вручну за допомогою Excel, Perfmon, Esxplot.

Veeam One – Має функцію моніторингу, складання технічних і бізнес-звітів. Ліцензується по процесорах, тому у випадку використанні «легких» VM, буде вигравати за ціною. Зручний і простий у керуванні. Є присутнім підтримка українською мовою. Широко відомий в Україні. – Ліцензується на процесор, у

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

випадку використанні “важких” ВМ, буде значно дорожче інших продуктів, які ліцензуються по кількості елементів моніторингу. Не підтримує фізичну інфраструктуру. Ні підтримки застосунків. – Працює з віртуалізацією від компанії VMware, надає метрики на основі власного збору інформації про продуктивність системи й дозволяє одержати інформацію, як для технічних фахівців, так і для людей, що ухвалюють рішення. – Дозволяє аналізувати найбільш важливі параметри. Не завантажує користувача непотрібною інформацією й налаштуваннями.

SpiceWorks – Безкоштовний. Дозволяє моніторити застосунки. – Працює з віртуальним середовищем лише непрямыми методами. Моніторинг ВМ ведеться з гостьовий ОС. У результаті відображає тільки параметри, характерні для фізичного середовища. Висновку про роботу гіпервізора й ВМ можна зробити, тільки порівнюючи параметри VSphere/VCenter з параметрами SpiceWorks. – Можна побачити характеристики фізичної інфраструктури, а також багато реклами. – Продукт створений для моніторингу фізичного середовища. Засоби аналізу віртуальної інфраструктури не розвинені. Можливо в подальших версіях.

Ми рекомендуємо: Veeam One

Застосунок для моніторингу 6-19 вузлів ESX

У таблиці нижче запропоновані самі підходящі варіанти.

При виборі ми враховували такі параметри, як: ціну, можливість, легкість у керуванні, а також підтримку в Україні.

Кількість об'єктів моніторингу розраховується, як якщо в одному ESX-хосте буде до 10 ВМ.

Для моніторингу й керування можна застосовувати вже більш складні й функціональні продукти. 6-19 вузлів ESX – це вже не мало, а в перекладі на об'єкти керування – це вже до 200.

Головними критеріями для таких систем будуть: можливість масштабування, потужний функціонал, але усе ще простий і зрозумілий інтерфейс.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Назва продукту – Переваги – Недоліки – Що можна побачити –
Процес моніторингу

Veeam One – Має функцію моніторингу, складання технічних і бізнес-звітів. Ліцензується по процесорах, тому у випадку використанні “легких” ВМ, буде вигравати за ціною. Зручний і простий у керуванні. Є присутнім підтримка українською мовою. Широко відомий в Україні. – Ліцензується на процесор, у випадку використанні “важких” ВМ, буде значно дорожче інших продуктів, які ліцензуються по кількості елементів моніторингу. Не підтримує фізичну інфраструктуру. Ні підтримки застосунків. – Працює з віртуалізацією від компанії VMware, надає метрики на основі власного збору інформації про продуктивність системи й дозволяє одержати інформацію, як для технічних фахівців, так і для людей, що ухвалюють рішення. – Дозволяє аналізувати найбільш важливі параметри. Не завантажує користувача непотрібною інформацією й налаштуваннями.

Orion VSP (Solarwinds) – Разом з Orion NPM моніторити фізичне середовище. Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і фізичного середовища здійснюється окремими продуктами. – Звіти моніторингу й аналізу для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунки), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Whats Virtual (Ipswitch) – Вбудовані засоби моніторингу й аналізу застосунків, графічна інформація надається в режимі реального часу. Підтримка

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Community. Разом з Whatsupgold можливо моніторити фізичне середовище. Моніторинг розподіленої інфраструктури з мінімально використовуваною смугою пропускання. – Не працює зі службою Active Directory. Відсутність підтримки українською мовою. Моніторинг фізичного середовища здійснюється окремим продуктом. – Видає інформацію із VM, ESX хостам, застосункам, а також має більш 200 готових шаблонів звітів. Графіки, інформація на яких видається в режимі реального часу. – Одержує дані з VCenter сервер. За допомогою графіків, що змінюються в режимі реального часу, спрощується моніторинг і аналіз інфраструктури.

VMware VCenter Operations Standard – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності. Не потрібно установки агентів на кожному із VM. Є представництво в Україні. – Для установки потрібні значні ресурси процесора й оперативної пам'яті. Не працює з застосунками. Немає функцій оповіщення. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). – Спрощує моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить VM і хости автоматично.

Ми рекомендуємо: Veeam One

Застосунок для моніторингу 20 і більш вузлів ESX

Отут виділені «гіганти» на ринку керування й моніторингу віртуальних систем.

Кожний із цих продуктів є Enterprise застосунком і підійде для моніторингу й керування необмеженого числа вузлів. Крім усього іншого, усі ці продукти мають величезний функціонал, а також працюють із величезним числом застосунків.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Головними критеріями в цьому «рецепті» ми виділили наступні: можливість швидкої оцінки всієї інфраструктури, орієнтованість на віртуальне середовище, спеціалізованість для кращого виконання вузького кола завдань по моніторингові й оптимізації віртуальної інфраструктури.

Назва продукту – Переваги – Недоліки – Що можна побачити – Процес моніторингу

Orion VSP (Solarwinds) – Разом з Orion NPM моніторити фізичне середовище. Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і фізичного середовища здійснюється окремими продуктами. – Звіти моніторингу й аналізу для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунок), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Zenoss Virtualization Monitoring – Автоматичний пошук (Auto discovery) усіх об'єктів моніторингу. Графічна вистава систем моніторингу з погляду бізнесу. Моніторинг хмарної структури. 3D вистава центру обробки даних. Не вимагає установки агентів на кожен із VM. Моніторинг застосунків уже включений у продукт. Для моніторингу фізичного середовища можна використовувати Server Monitoring. – Не розрахований на малі й середні підприємства, мінімальний пакет для моніторингу – 200 об'єктів (VM, ESX хости, застосунки). Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг фізичного середовища здійснюється окремим продуктом. – Карти, що відображають у режимі реального часу всі філії компанії і їх статус.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Бізнес інформацію. Створення планів серверної, показує сервера і їх статус. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – Технологія авто виявлення дозволяє додати об'єкти в систему моніторингу автоматично. Самостійно класифікує об'єкти й будує їхню топологію.

VMware VCenter Operations Enterprise – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності, а також пропонує застосунок для проблемних об'єктів. Не потрібно установки агентів на кожен із ВМ. Є представництво в Україні. Пропонує значення параметрів для Limits і Reservations. Робить моніторинг AD. Спеціально розроблений для моніторингу віртуального середовища. – Крім AD, не моніторити застосунки. Немає моніторингу фізичного середовища. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). Також можна побачити пропозиції по усуненню проблем з об'єктами. – Спрощує моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить ВМ і хости автоматично.

Ми рекомендуємо: VCenteroperations Enterprise

Безкоштовні засоби моніторингу

Тут представлені найкращі (на нашу думку) з безкоштовних засобів моніторингу.

При рекомендації одного із цих продуктів ми відштовхувалися від того, що надалі, при росту числа вузлів ESX, Ви можете безболісно перейти на продукт тієї ж компанії, але вже платний і з більшим функціоналом.

Критерії були такі: продукт повинен бути безкоштовним, уміти надавати осудну інформацію про стан інфраструктури, мати можливість подальшого апгрейда на повнофункціональний продукт.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Назва продукту – Переваги – Недоліки – Що можна побачити –
Процес моніторингу

Management Assistant – Веде автоматизований моніторинг мережі. – Споконвічно працює автоматизоване тільки з 1 хостом, для того щоб велися логи всіх хостів, прийде писати численні скрипти. – Оскільки vMA є засобом збору статистики, на виході ми отримуємо .csv файл із численними даними про сервер ESX і VM на ньому. – vMA погано придатний для аналізу, оскільки аналіз повинен здійснюватися вручну за допомогою Excel, Perfmon, Esxplot.

Veeam Monitor Free Edition – Можливість перейти в будь-який момент на повноцінну версію Veeam Monitor. – Без бандла Veeam One має порізаний функціонал – Видає інформацію про продуктивність серверів і VM, показує проблемні зони, а також компресію сервера і його своп. – Якщо ми вибрали безкоштовну технологію, то продукт може використовуватися в тестових цілях, або показово для рекламування продукту Veeam One. Платна ж версія велике й зручно відображає всі потрібні нам відомості для аналізу інфраструктури

SpiceWorks – Дозволяє моніторити застосунки. – Працює з віртуальним середовищем лише непрямыми методами. Моніторинг VM ведеться з гостьовий ОС. У результаті відображає тільки параметри, характерні для фізичного середовища. Висновку про роботу гіпервізора й VM можна зробити, тільки порівнюючи параметри VSphere/VCenter з параметрами SpiceWorks. – Можна побачити характеристики фізичної інфраструктури, а також багато реклами. – Продукт створений для моніторингу фізичного середовища. Засоби аналізу віртуальної інфраструктури не розвинені. Можливо в подальших версіях.

Ми рекомендуємо: Veeam Monitor Free Edition

Застосунок для моніторингу розподіленої мережі вузлів

У цьому «рецепті» ми спробували врахувати інтереси компаній із сильно розподіленою інфраструктурою.

Десятки, сотні, тисячі філій, офісів, точок присутності, що географічно відстоять друг від друга на десятки, сотні й тисячі кілометрів.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Головними критеріями в оцінці системи моніторингу в цьому випадку стали: гнучкий рівень деталізації інфраструктури (від перегляду інформації з усього світу, до конкретного сервера), кинувши погляд на карту об'єктів моніторингу адміністратор повинен розуміти в якій країні, у якому місті виникла проблема, а потім уже спускатися на рівень об'єкта й усувати неполадки. Функціонал такої системи повинен бути дуже потужним.

Назва продукту – Переваги – Недоліки – Що можна побачити – Процес моніторингу

Orion VSP (Solarwinds) – Разом з Orion NPM моніторити фізичне середовище. Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і фізичного середовища здійснюється окремими продуктами. – Звіти моніторингу й аналізу для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунок), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Zenoss Virtualization Monitoring – Автоматичний пошук (Auto discovery) усіх об'єктів моніторингу. Графічна вистава систем моніторингу з погляду бізнесу. Моніторинг хмарної структури. 3D вистава центру обробки даних. Не вимагає установки агентів на кожен із VM. Моніторинг застосунків уже включений у продукт. Для моніторингу фізичного середовища можна використовувати Server Monitoring. – Не розрахований на малі й середні підприємства, мінімальний пакет для моніторингу – 200 об'єктів (VM, ESX хости, застосунки). Відсутність підтримки українською мовою, немає представництва в

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Україні. Моніторинг фізичного середовища здійснюється окремим продуктом. – Карти, що відображають у режимі реального часу всі філії компанії і їх статус. Бізнес інформацію. Створення планів серверної, показує сервера і їх статус. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – Технологія авто виявлення дозволяє додати об'єкти в систему моніторингу автоматично. Самостійно класифікує об'єкти й будує їхню топологію.

Whats Virtual (Ipswitch) – Вбудовані засоби моніторингу й аналізу застосунків, графічна інформація надається в режимі реального часу. Підтримка Community. Разом з Whatsupgold можливо моніторити фізичне середовище. Моніторинг розподіленої інфраструктури з мінімально використовуваною смугою пропускання. – Не працює зі службою Active Directory. Відсутність підтримки українською мовою. Моніторинг фізичного середовища здійснюється окремим продуктом. – Видає інформацію із VM, ESX хостам, застосункам, а також має більш 200 готових шаблонів звітів. Графіки, інформація на яких видається в режимі реального часу. – Одержує дані з VCenter сервер. За допомогою графіків, що змінюються в режимі реального часу, спрощується моніторинг і аналіз інфраструктури.

VMware VCenter Operations Enterprise – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності, а також пропонує застосунок для проблемних об'єктів. Не потрібно установки агентів на кожен із VM. Є представництво в Україні. Пропонує значення параметрів для Limits і Reservations. Робить моніторинг AD. Спеціально розроблений для моніторингу віртуального середовища. – Крім AD, не моніторити застосунки. Немає моніторингу фізичного середовища. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). Також можна побачити пропозиції по усуненню проблем з об'єктами. – Спрощує моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить ВМ і хости автоматично.

Ми рекомендуємо: Orion

Засоби моніторингу із пропозиціями по оптимізації

Тут представлений перелік продуктів, що не тільки надають звіти моніторингу, але застосунки, що й пропонують свої, по усуненню яких-небудь проблемних зон.

Система моніторингу повинна добре «розбиратися» у причинах виникнення проблем і пропонувати способи їх усунення.

Ми не вважаємо правильним повністю покладатися на підказки системи, однак, при виникненні типових проблем, швидко усувати їх може адміністратор не із самою вищою кваліфікацією.

Можливо, це допоможе оптимізувати витрати на першу лінію технічної підтримки.

Хоча, у системах моніторингу все найвищою мірою персонально.

Головний критерій вибору системи керування – наявність «інтелекту» у системи керування.

Назва продукту – Переваги – Недоліки – Що можна побачити – Процес моніторингу

Orion VSP (Solarwinds) – Разом з Orion NPM моніторити фізичне середовище. Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і фізичного середовища здійснюється окремими продуктами. – Звіти моніторингу й аналізу для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунку), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Zenoss Virtualization Monitoring – Автоматичний пошук (Auto discovery) усіх об'єктів моніторингу. Графічна вистава систем моніторингу з погляду бізнесу. Моніторинг хмарної структури. 3D вистава центру обробки даних. Не вимагає установки агентів на кожному із VM. Моніторинг застосунків уже включений у продукт. Для моніторингу фізичного середовища можна використовувати Server Monitoring. – Не розрахований на малі й середні підприємства, мінімальний пакет для моніторингу – 200 об'єктів (VM, ESX хости, застосунки). Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг фізичного середовища здійснюється окремим продуктом. – Карти, що відображають у режимі реального часу всі філії компанії і їх статус. Бізнес інформацію. Створення планів серверної, показує сервера і їх статус. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – Технологія авто виявлення дозволяє додати об'єкти в систему моніторингу автоматично. Самостійно класифікує об'єкти й будує їхню топологію.

VMware VCenter Operations Enterprise – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності, а також пропонує застосунку для проблемних об'єктів. Не потрібно установки агентів на кожному із VM. Є представництво в Україні. Пропонує значення параметрів для Limits і Reservations. Робить моніторинг AD. Спеціально розроблений для моніторингу віртуального середовища. – Крім AD, не моніторити застосунки. Немає моніторингу фізичного середовища. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). Також можна побачити пропозиції по усуненню проблем з об'єктами. – Спрощує

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить ВМ і хости автоматично.

Ми рекомендуємо: VCenteroperations Enterprise

Засоби швидкого реагування

Важливо одержувати інформацію про стан системи швидко й у будь-який час.

Адміністратор повинен бути сповіщений ще до того, як трапилося непоправне, тому ми вважаємо, що «засоби швидкого реагування» доречні для компаній, що прагнуть надавати якісні послуги як зовнішнім, так і внутрішнім клієнтам.

Головними критеріями є: швидкість реакції й гнучкість налаштування повідомлень, а також спеціалізованість засобів. Адміністратор віртуальної інфраструктури повинен бути швидко введений у курс того, що відбулося або може відбутися з інфраструктурою й одержати рекомендацію з усунення несправності.

Оскільки ми прагнемо одержати рекомендації з усунення, продукт моніторингу повинен як мінімум це вміти, а як максимум виявляти функціональну область, де перебуває причина проблеми й сповіщати співробітників, відповідальних за цю функціональну область.

Назва продукту – Переваги – Недоліки – Що можна побачити – Процес моніторингу

Orion VSP (Solarwinds) – Разом з Orion NPM моніторити фізичне середовище. Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і фізичного середовища здійснюється окремими продуктами. – Звіти моніторингу й аналізу

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунку), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Zenoss Virtualization Monitoring – Автоматичний пошук (Auto discovery) усіх об'єктів моніторингу. Графічна вистава систем моніторингу з погляду бізнесу. Моніторинг хмарної структури. 3D вистава центру обробки даних. Не вимагає установки агентів на кожен із VM. Моніторинг застосунків уже включений у продукт. Для моніторингу фізичного середовища можна використовувати Server Monitoring. – Не розрахований на малі й середні підприємства, мінімальний пакет для моніторингу – 200 об'єктів (VM, ESX хости, застосунки). Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг фізичного середовища здійснюється окремим продуктом. – Карти, що відображають у режимі реального часу всі філії компанії і їх статус. Бізнес інформацію. Створення планів серверної, показує сервера і їх статус. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – Технологія авто виявлення дозволяє додати об'єкти в систему моніторингу автоматично. Самостійно класифікує об'єкти й будує їхню топологію.

Whats Virtual (Ipswitch) – Вбудовані засоби моніторингу й аналізу застосунків, графічна інформація надається в режимі реального часу. Підтримка Community. Разом з Whatsupgold можливо моніторити фізичне середовище. Моніторинг розподіленої інфраструктури з мінімально використовуваною смугою пропускання. – Не працює зі службою Active Directory. Відсутність підтримки українською мовою. Моніторинг фізичного середовища здійснюється окремим продуктом. – Видає інформацію із VM, ESX хостам, застосункам, а

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

також має більш 200 готових шаблонів звітів. Графіки, інформація на яких видається в режимі реального часу. – Одержує дані з VCenter сервер. За допомогою графіків, що змінюються в режимі реального часу, спрощується моніторинг і аналіз інфраструктури.

VMware VCenter Operations Enterprise – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності, а також пропонує застосунку для проблемних об'єктів. Не потрібно установки агентів на кожен із ВМ. Є представництво в Україні. Пропонує значення параметрів для Limits і Reservations. Робить моніторинг AD. Спеціально розроблений для моніторингу віртуального середовища. – Крім AD, не моніторити застосунки. Немає моніторингу фізичного середовища. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). Також можна побачити пропозиції по усуненню проблем з об'єктами. – Спрощує моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить ВМ і хости автоматично.

Ми рекомендуємо: VCenteroperations Enterprise

Засоби, що підходять як для віртуального середовища, так і для фізичного середовища

Можливо, у фізичній інфраструктурі вже використовувалися засоби моніторингу й керування.

При переносі частини потужностей у віртуальну інфраструктуру встає питання керування і їй.

Ми спробували оцінити, як діяти в цьому випадку.

Головними критеріями служила наявність системи керування фізичною інфраструктурою й можливість інтеграції систем моніторингу фізичної й віртуальної інфраструктури.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Назва продукту – Переваги – Недоліки – Що можна побачити –
Процес моніторингу

Orion VSP (Solarwinds) – Разом з Orion NPM моніторити фізичне середовище. Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і фізичного середовища здійснюється окремими продуктами. – Звіти моніторингу й аналізу для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунок), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Zenoss Virtualization Monitoring – Автоматичний пошук (Auto discovery) усіх об'єктів моніторингу. Графічна вистава систем моніторингу з погляду бізнесу. Моніторинг хмарної структури. 3D вистава центру обробки даних. Не вимагає установки агентів на кожен із VM. Моніторинг застосунків уже включений у продукт. Для моніторингу фізичного середовища можна використовувати Server Monitoring. – Не розрахований на малі й середні підприємства, мінімальний пакет для моніторингу – 200 об'єктів (VM, ESX хости, застосунки). Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг фізичного середовища здійснюється окремим продуктом. – Карти, що відображають у режимі реального часу всі філії компанії і їх статус. Бізнес інформацію. Створення планів серверної, показує сервера і їх статус. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – Технологія авто виявлення дозволяє додати об'єкти в систему моніторингу автоматично. Самостійно класифікує об'єкти й будує їхню топологію.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Whats Virtual (Ipswitch) – Вбудовані засоби моніторингу й аналізу застосунків, графічна інформація надається в режимі реального часу. Підтримка Community. Разом з Whatsupgold можливо моніторити фізичне середовище. Моніторинг розподіленої інфраструктури з мінімально використовуваною смугою пропускання. – Не працює зі службою Active Directory. Відсутність підтримки українською мовою. Моніторинг фізичного середовища здійснюється окремим продуктом. – Видає інформацію із VM, ESX хостам, застосункам, а також має більш 200 готових шаблонів звітів. Графіки, інформація на яких видається в режимі реального часу. – Одержує дані з VCenter сервер. За допомогою графіків, що змінюються в режимі реального часу, спрощується моніторинг і аналіз інфраструктури.

VMware VCenter Operations Enterprise – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності, а також пропонує застосунок для проблемних об'єктів. Не потрібно установки агентів на кожному із VM. Є представництво в Україні. Пропонує значення параметрів для Limits і Reservations. Робить моніторинг AD. Спеціально розроблений для моніторингу віртуального середовища. – Крім AD, не моніторити застосунки. Немає моніторингу фізичного середовища. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). Також можна побачити пропозиції по усуненню проблем з об'єктами. – Спрощує моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить VM і хости автоматично.

Veeam nworks Management Pack 5.6 для MS SCOM – Інтегрується з наявною системою SCOM. Додає в SCOM можливість моніторингу віртуального середовища. Є підтримка українською мовою й представництво в Україні. – Не має змісту без MS SCOM. – За допомогою плагіна ми можемо побачити

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

віртуальну інфраструктуру VMware в інтерфейсі SCOM. – Може бути вбудований як плагін в MS SCOM, так і в Virtual Machine Manager, тому що Veeam буде збирати інформацію із усіх параметрів, використовуваними VM і ESX.

Veeam nworks Smart Plug-In 5.6 для HP OM – Інтегрується з наявною системою SCOM. Додає в HP OM. можливість моніторингу віртуального середовища. Є підтримка українською мовою й представництво в Україні. – Не має змісту без HP OM. – За допомогою плагіна ми можемо побачити віртуальну інфраструктуру в інтерфейсі HP OM. – Дуже схожий на рідний продукт HP Smart Plug-in, але у відмінності від нього не підтримуючий технологію Auto Discovery.

Ми рекомендуємо:

Якщо є присутнім один із продуктів MS SCOM або HP OM, варто вибрати плагіни компанії Veeam:

- Veeam nworks Management Pack 5.6 для Microsoft System Center Operation Manager.
- Veeam nworks Smart Plug-In 5.6 для HP Operation Manager.

У випадку якщо ми вже користувалися системами моніторингу від компаній Ipswitch, Solarwinds, Zenoss, слід вибрати їхній застосунок для моніторингу віртуального середовища.

Якщо до цього ми не використовували системи моніторингу у фізичному середовищі, при переході на віртуальну інфраструктуру слід вибрати VCenter Operations.

Керування продуктивністю й моніторинг застосунків

Залежно від розміру й розвиненості інфраструктури, може бути необхідно моніторити застосунки, що працюють на віртуальних машинах.

Нижче ми представили перелік продуктів моніторингу, що більш-менш успішно працюють із застосунками й спрямованих на моніторинг інфраструктури.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

У статті порівняння засобів моніторингу застосунків Ви зможете знайти повний перелік застосунків, підтримуваних кожним з наведених продуктів моніторингу.

Оскільки в кожному конкретному випадку використовуються свої додатки, ми просимо Вас оцінити, які з наведених застосунків найбільш важливі саме для Вас.

Ми ж за головний критерій брали найбільша кількість застосунків, підтримуваних продуктом моніторингу.

Назва продукту – Переваги – Недоліки – Що можна побачити – Процес моніторингу

Orion VSP (Solarwinds) – Разом із продуктом Orion APM може моніторити застосунки (Exchange, AD, SQL і т.д.). Є ліцензія на необмежену кількість елементів моніторингу. Підтримка Community. – Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг застосунків і здійснюється окремим продуктом. – Звіти моніторингу й аналізу для віртуальної інфраструктури (і ESX, і VM), для моніторингу фізичних серверів або застосунків необхідно придбати окремі продукти. Характеристики й робота застосунків. Характеристики й роботи фізичних серверів. – В автоматичному режимі аналізує віртуальну інфраструктуру (а також фізичну інфраструктуру й застосунку), гнучке налаштування одержання попереджень. Є функція сканування сервера VCenter і одержання інформації про всій інфраструктурі для простоти додавання ESX хостів.

Zenoss Virtualization Monitoring – Автоматичний пошук (Auto discovery) усіх об'єктів моніторингу. Графічна вистава систем моніторингу з погляду бізнесу. Моніторинг хмарної структури. 3D вистава центру обробки даних. Не вимагає установки агентів на кожен із VM. Моніторинг застосунків уже включений у продукт. Для моніторингу фізичного середовища можна використовувати Server Monitoring. – Не розрахований на малі й середні підприємства, мінімальний пакет для моніторингу – 200 об'єктів (VM, ESX хости,

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

застосунки). Відсутність підтримки українською мовою, немає представництва в Україні. Моніторинг фізичного середовища здійснюється окремим продуктом. – Карти, що відображають у режимі реального часу всі філії компанії і їх статус. Бізнес інформацію. Створення планів серверної, показує сервера і їх статус. Характеристики й робота застосунків (посилання на перелік). Характеристики й роботи фізичних серверів. – Технологія автознаходження дозволяє додати об'єкти в систему моніторингу автоматично. Самостійно класифікує об'єкти й будує їхню топологію.

VMware VCenter Operations Enterprise – Привласнює кожному об'єкту рейтинг продуктивності. Інтегрується в VSphere Client. Створює звіти по продуктивності, а також пропонує застосунки для проблемних об'єктів. Не потрібно установки агентів на кожен із ВМ. Є представництво в Україні. Пропонує значення параметрів для Limits і Reservations. Робить моніторинг AD. Спеціально розроблений для моніторингу віртуального середовища. – Крім AD, не моніторити застосунки. Немає моніторингу фізичного середовища. – На підставі рейтингу продуктивності кожному об'єкту відповідає колір від зеленого (об'єкт у порядку) до червоного (об'єкт вимагає термінового втручання). Також можна побачити пропозиції по усуненню проблем з об'єктами. – Спрощує моніторинг і аналіз застосунків, оскільки всі дії будуть відбуватися в централізованому інтерфейсі (VSphere Client). Оскільки ставиться плагіном, немає необхідності “пояснювати” йому нашу інфраструктуру, сканує систему й знаходить ВМ і хости автоматично.

Whatsvirtual (Ipswitch) – Вбудовані засоби моніторингу й аналізу застосунків, графічна інформація надається в режимі реального часу. Підтримка Community. Разом з Whatsupgold можливо моніторити фізичне середовище. Моніторинг розподіленої інфраструктури з мінімально використовуваною смугою пропускання. – Не працює зі службою Active Directory. Відсутність підтримки українською мовою. Моніторинг фізичного середовища здійснюється окремим продуктом. – Видає інформацію із ВМ, ESX хостам, застосункам, а

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

також має більш 200 готових шаблонів звітів. Графіки, інформація на яких видається в режимі реального часу. – Одержує дані з VCenter сервер. За допомогою графіків, що змінюються в режимі реального часу, спрощується моніторинг і аналіз інфраструктури.

Ми рекомендуємо: Zenoss

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – динамічна інтерпретована об’єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

Використання Python

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.

Версії Python

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з'являються різні версії

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп'ютері. Одна з основних функцій процесора – це обробка даних згідно комп'ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп'ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп'ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Декодувати

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

Виконати

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

Середовище програмування для Python

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

- IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.
- Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.
- Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.
- PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.
- Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.
- Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

– Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

– Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи аналізу трафіку віртуальної інфраструктури.

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) сформулювати висновки про виконаний обсяг робіт та одержані результати.

К6ПЗ-2026

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Як відомо, в VMware VSphere 5 з'явилося кілька корисних нововведень, що стосуються мережної взаємодії, доступних у розподіленому комутаторі VMware VSphere Distributed Switch (vds), які полегшують життя мережним адміністраторам. Зокрема, за допомогою dvswitch доступні наступні нові можливості, які описані в документі "What's New in VMware VSphere 5.0 Networking":

- Підтримка Netflow версії 5 – можливість перегляду трафіку між віртуальними машинами (ВМ-ВМ на одному або різних хостах, а також ВМ-фізичний сервер) за допомогою сторонніх продуктів, що підтримують Netflow.

- Підтримка дзеркалювання портів Switch Port Analyzer (аналог технології SPAN у комутаторах Cisco) – можливість дублювати трафік віртуальної машини (а також Vmkernel і фізичних адаптерів) на цільову машину (Port Mirroring), яка може реалізовувати функціонал системи виявлення або запобігання вторгнень (IDS/IPS).

- Підтримка відкритого стандарту Link Layer Discovery Protocol (LLDP, у реалізації 802.1AB) – це механізм виявлення сусідніх мережних пристроїв і збору інформації про них для вирішення різних проблем мережними адміністраторами. Раніше підтримувався тільки протокол CDP (Cisco Discovery Protocol), підтримка якого є не у всіх пристроях.

- Поліпшення механізму Network I/O Control – пули ресурсів для мережного трафіку й підтримка стандарту 802.1q. Визначає користувачем пули для різних типів трафіку дозволяють пріоритезувати і обмежувати пропускну здатність каналу для них за допомогою механізму shares і limits.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Усі ці нові можливості ми розберемо в наступних замітках, а сьогодні зосередимося на механізмі Netflow і його підтримці в VSphere 5. Netflow – мережний протокол, призначений для обліку мережного трафіку, розроблений компанією Cisco Systems. Є фактичним промисловим стандартом і підтримується не тільки встаткуванням Cisco, але й багатьма іншими пристроями.

Для збору інформації про трафік по протоколу Netflow потрібні наступні компоненти:

- Сенсор. Збирає статистику по минаючому через нього трафіку. Звичайно це L 3-комутатор або маршрутизатор, хоча можна використовувати й окремо варті сенсори, що одержують дані шляхом дзеркалювання порту комутатора. У нашому випадку це розподілений комутатор vds.
- Колектор. Збирає одержувані від сенсора дані й поміщає їх у сховище.
- Аналізатор. Аналізує зібрані колектором дані й формує придатні для читання людиною звіти (часто у вигляді графіків).

Netflow дає можливість мережному адміністраторові моніторити мережні взаємодії віртуальних машин для подальших дій по виявленню мережних вторгнень, відстеження відповідності конфігурацій мережних служб і аналізу в цілому. Крім того, дана можливість корисна тоді, коли потрібно відслідковувати потік трафіку від застосунків усередині віртуальної машини з метою контролю продуктивності мережі й цільового використання трафіку.

Налаштований віртуальний комутатор, посилає дані Netflow на сторонню машину (колектор), яка підключена до хост-серверу VMware ESXi через фізичний комутатор. Колектор уже передає дані аналізатору. Netflow може бути включений на рівні окремої групи портів (dvportgroup), окремого порту або аплінка (Uplink).

Для початку налаштування Netflow потрібно зайти у властивості комутатора vds (він повинен бути версії 5.0.0 або вище).

Тут ми вказуємо IP-адресу колектора, куди будуть відправлятися дані, його порт, а також єдиний IP-адреса комутатора vds, щоб хости не представлялися окремими комутаторами для колектора.

Включити моніторинг Netflow можна у властивостях групи портів на vds у розділі Monitoring.

Далі в цю групу портів включаємо одну з віртуальних машин.

Тепер можна використовувати один із продуктів для збору й аналізу трафіку Netflow, наприклад, Manage Engine Netflow Analyzer. Приклад статистики, яку збирає цей продукт по протоколах (у цьому випадку більшість трафіку – http):

Netflow можна використовувати для різних цілей моніторингу, наприклад, в інфраструктурі VMware View, де присутні сотні віртуальних машин, можна згрупувати трафік по групах і дивитися, скільки трафіку виживається відеосервісами (Youtube, наприклад), тому що це може сильно впливати на продуктивність мережі в цілому:

Застосувань Netflow насправді безліч, тому його підтримка в VMware VSphere 5 може виявитися вам дуже корисною.

3.2 Розробка структурної схеми

Віртуалізація міцно увійшла в наше життя, і ми все частіше в проектах зустрічаємо необхідність захвату трафіку або зняття копії трафіку для моніторингу продуктивності застосунків, які розгорнуті усередині віртуальної інфраструктури. Компанія VMWARE представила й активно розбудовує нову платформу NSX для віртуалізації мережі, яка набирає популярність. Тому в рамках даного розділу розглянемо, як розроблене в даній роботі програмне забезпечення VirtualInfrastructureTrafficAnalysis можна використовувати для діагностики NSX і захвату трафіку.

Для того щоб оцінити можливості був зібраний стенд, який зображено на рисунку 3.1. Для кожного експерименту генерувався трафік ICMP і/або SSH між віртуальними машинами:

Ну а тепер приступимося до захвату трафіку в різних місцях віртуального простору.

1. Захват трафіку з і у віртуальну машину

Для того щоб контролювати трафік на рівні входу й виходу з віртуальної машини ми можемо скористатися ідентифікатором порту віртуального комутатора –switchport . У даній точці ми бачимо трафік ще не порушений правилами NSX файєрволу й не інкапсульований у який-небудь VXLAN. У даному прикладі ми будемо збирати трафік при спілкуванні між двома віртуальними машинами Web 1 і Web 2 на рівні віртуального мережного адаптера Web1.

Для того щоб одержати ідентифікатор порту (port-Id) існують два способи. Перший спосіб запустити команду esxtop на ESXi Compute1 і одержати список, у якому є присутнім і наш сервер Web 1.

Спосіб 2 запустити команду net-stats -l і також побачити port-Id:

```
[root@compute1:~] net-stats -l
Portnum Type Subtype Switchname Macaddress Clientname
33554434 4 0 vswitch0 00:0c:29:db:8c:f0 vmnic0
33554436 3 0 vswitch0 00:0c:29:db:8c:f0 vmk0
33554437 3 0 vswitch0 00:50:56:63:1b:bb vmk1
```

33554438 3 0 vswitch0 00:50:56:65:b8:b8 vmk3 Як відомо, в VMware VSphere 5 з'явилося кілька корисних нововведень, що стосуються мережної взаємодії, доступних у розподіленому комутаторі VMware VSphere Distributed Switch (vds), які полегшують життя мережним адміністраторам. Зокрема, за допомогою dvswitch доступні наступні нові можливості, які описані в документі "What's New in VMware VSphere 5.0 Networking":

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

– Підтримка Netflow версії 5 – можливість перегляду трафіку між віртуальними машинами (ВМ-ВМ на одному або різних хостах, а також ВМ-фізичний сервер) за допомогою сторонніх продуктів, що підтримують Netflow.

– Підтримка дзеркалювання портів Switch Port Analyzer (аналог технології SPAN у комутаторах Cisco) – можливість дублювати трафік віртуальної машини (а також Vmkernel і фізичних адаптерів) на цільову машину (Port Mirroring), яка може реалізовувати функціонал системи виявлення або запобігання вторгнень (IDS/IPS).

– Підтримка відкритого стандарту Link Layer Discovery Protocol (LLDP, у реалізації 802.1AB) – це механізм виявлення сусідніх мережних пристроїв і збору інформації про них для вирішення різних проблем мережними адміністраторами. Раніше підтримувався тільки протокол CDP (Cisco Discovery Protocol), підтримка якого є не у всіх пристроях.

– Поліпшення механізму Network I/O Control – пули ресурсів для мережного трафіку й підтримка стандарту 802.1q. Визначає користувачем пули для різних типів трафіку дозволяють пріоритетизувати і обмежувати пропускну здатність каналу для них за допомогою механізму shares і limits.

Зупинимося на роботі протоколу Netflow у середовищі vSphere 5. Netflow – це розроблений Cisco Systems стандарт для обліку мережевого трафіку, який сьогодні підтримується більшістю виробників мережевого обладнання.

Для функціонування системи моніторингу на базі Netflow необхідні три ключові елементи:

– Сенсор: джерело статистики (маршрутизатор, L3-комутатор або, як у нашому випадку, розподілений віртуальний комутатор vDS).

– Колектор: сервер, що приймає дані від сенсора та зберігає їх у базі.

– Аналізатор: програмний інструмент, який обробляє накопичену інформацію та візуалізує її у вигляді звітів і графіків.

Netflow дає можливість мережному адміністраторові моніторити мережні взаємодії віртуальних машин для подальших дій по виявленню мережних

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

вторгнень, відстеження відповідності конфігурацій мережних служб і аналізу в цілому. Крім того, дана можливість корисна тоді, коли потрібно відслідковувати потік трафіку від застосунків усередині віртуальної машини з метою контролю продуктивності мережі й цільового використання трафіку.

Налаштований віртуальний комутатор, посилає дані Netflow на сторонню машину (колектор), яка підключена до хост-серверу VMware ESXi через фізичний комутатор. Колектор уже передає дані аналізатору. Netflow може бути включений на рівні окремої групи портів (dvportgroup), окремого порту або аплінка (Uplink).

Для початку налаштування Netflow потрібно зайти у властивості комутатора vds (він повинен бути версії 5.0.0 або вище).

Тут ми вказуємо IP-адресу колектора, куди будуть відправлятися дані, його порт, а також єдиний IP-адреса комутатора vds, щоб хости не представлялися окремими комутаторами для колектора.

Включити моніторинг Netflow можна у властивостях групи портів на vds у розділі Monitoring.

Далі в цю групу портів включаємо одну з віртуальних машин.

Тепер можна використовувати один із продуктів для збору й аналізу трафіку Netflow, наприклад, Manage Engine Netflow Analyzer. Приклад статистики, яку збирає цей продукт по протоколах (у цьому випадку більшість трафіку – http):

Netflow можна використовувати для різних цілей моніторингу, наприклад, в інфраструктурі VMware View, де присутні сотні віртуальних машин, можна згрупувати трафік по групах і дивитися, скільки трафіку виживається відеосервісами (Youtube, наприклад), тому що це може сильно впливати на продуктивність мережі в цілому:

Застосувань Netflow насправді безліч, тому його підтримка в VMware VSphere 5 може виявитися вам дуже корисною.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

3.2 Розробка структурної схеми

Віртуалізація міцно увійшла в наше життя, і ми все частіше в проектах зустрічаємо необхідність захвату трафіку або зняття копії трафіку для моніторингу продуктивності застосунків, які розгорнуті усередині віртуальної інфраструктури. Компанія VMWARE представила й активно розбудовує нову платформу NSX для віртуалізації мережі, яка набирає популярність. Тому в рамках даного розділу розглянемо, як розроблене в даній роботі програмне забезпечення VirtualInfrastructureTrafficAnalysis можна використовувати для діагностики NSX і захвату трафіку.

Для того щоб оцінити можливості був зібраний стенд, який зображено на рисунку 3.1. Для кожного експерименту генерувався трафік ICMP і/або SSH між віртуальними машинами:

Ну а тепер приступимося до захвату трафіку в різних місцях віртуального простору.

1. Захват трафіку з і у віртуальну машину

Для того щоб контролювати трафік на рівні входу й виходу з віртуальної машини ми можемо скористатися ідентифікатором порту віртуального комутатора –switchport . У даній точці ми бачимо трафік ще не порушений правилами NSX файєрволу й не інкапсульований у який-небудь VXLAN. У даному прикладі ми будемо збирати трафік при спілкуванні між двома віртуальними машинами Web 1 і Web 2 на рівні віртуального мережного адаптера Web1.

Для того щоб одержати ідентифікатор порту (port-Id) існують два способи. Перший спосіб запустити команду esxtop на ESXi Compute1 і одержати список, у якому є присутнім і наш сервер Web 1.

Спосіб 2 запустити команду net-stats -l і також побачити port-Id:

Для того щоб бачити трафік у двох напрямках нам необхідно запустити 2 сесії захвату трафіку – для напрямку з машини Web1 – це напрямок –dir 0, а

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

напрямок у машину Web 1 – dir 1. Формат і опції команди VirtualInfrastructureTrafficAnalysis можна уточнити в базі знань VMware.

Між машинами генерувався трафік і всі пакети зберігалися у файли VM-WEB1.OUT.pcap і VM-WEB1-IN.pcap.

Далі можна скористатися VirtualInfrastructureTrafficAnalysis для аналізу трафіку.

Як видно із прикладу ICMP трафік пройшов від машини Web 1 до машини Web 2 і отримана відповідь. А от трафік SSH не пройшов і отут спрацювали правила на фаєрволе NSX DFW (пам'ятаємо вище, що ми дозволили прохід тільки ICMP трафіку).

2. Захват трафіку після NSX Distributed Firewall (DFW)

У процесі діагностики може знадобитися перевірити роботу NSX DFW і тому розглянемо наступну точку після фільтрів фаєрвола. Для цього існують дві опції для VirtualInfrastructureTrafficAnalysis:

-capture <точка захвату>

Таблиця 3.1 – Варіанти й застосування двох точок захвату:

Точка захвату	Використовується для
Predvfilter	захвату пакетів перед тем як фільтр буде застосований.
Postdvfilter	захвату пакетів після фільтрації.

Для того щоб трафік захопити в точці 2 нам потрібна опція –capture Postdvfilter.

Dvfilter який ми прагнемо застосувати для конкретної віртуальної мережної карти vnic

У даному експерименті ми будемо захоплювати трафік від машини Web 1 у точці Postdvfilter. Але спочатку ми повинні довідатися ім'я фільтра для карти Web 1. Це може бути зроблене, використовуючи команду summarize-dvfilter.

Далі ми можемо знову запустити захват трафіку (ICMP і SSH).

Подивимося наш файл WEB1-POSTDVFILTER.pcap.

Як ми бачимо, трафік ICMP пройшов без проблем, а SSH трафіку не видне, тому що його не пропустив файєрвол.

3. Захват інкапсульованого трафіку VXLAN

У даному прикладі ми будемо захоплювати трафік після його інкапсуляції. Наприклад, для вирішення проблем зв'язки віртуального комутатора з фізичним мережним адаптером на фізичному сервері. Для застосунку цього завдання використовуються наступні опції для VirtualInfrastructureTrafficAnalysis:

-capture <точка захвату>

Таблиця 3.2 – Точка захвату як завжди залежить від напрямку трафіку:

Точка захвату	Використовується для
Uplinksnd	захвату пакетів у напрямку від віртуального комутатора убік фізичного мережного адаптера
Uplinkrcv	захвату пакетів від фізичного мережного адаптера

-uplink <vmmicx>

Ім'я фізичного мережного адаптера (vmmic)

-vxlan <VNI>

Номер VXLAN

У даному прикладі ми захоплюємо VXLAN трафік для VNI 5001.

Під час захвату виконали команду ping від Web1 убік Web2.

4. Захват трафіку після маршрутизатора NSX Distributed Logical Router (DLR)

Якщо необхідно перевірити коректність роботи вбудованого розподіленого логічного маршрутизатора, то для застосунку цього завдання необхідно з'ясувати ідентифікатор порту port-Id маршрутизатора vdrport. Для цього ми можемо скористатися утилітою ESXTOP або через командний рядок NSX.

Для визначення напрямку трафіку ми використовуємо опції -dir 0 для захвату трафіку від DLR (egress) убік віртуального комутатора. Крім цього ми можемо вказати інтерфейс із вихідним трафіком, використовуючи -vxlan <VNI>.

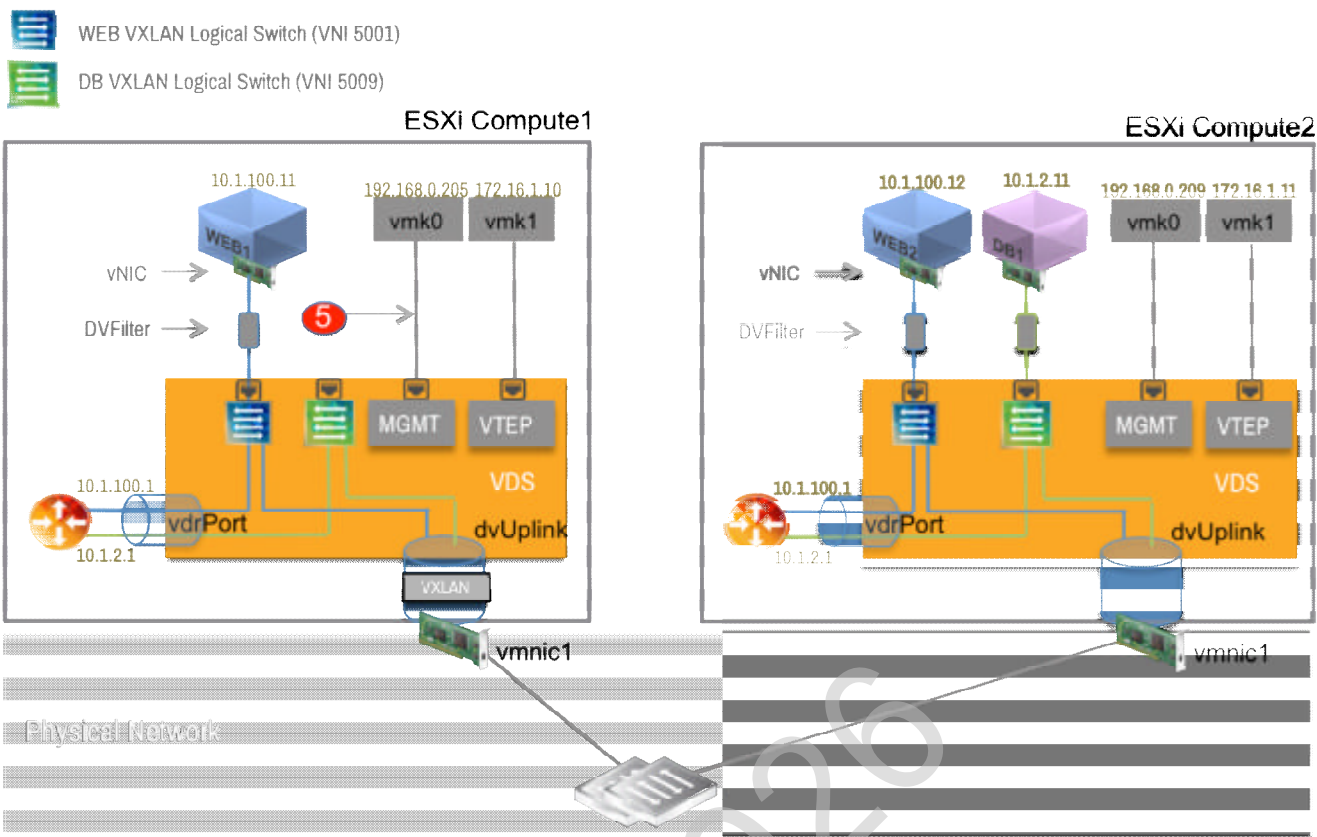


Рисунок 3.1 – Структурна схема системи

На прикладі нижче ми використовуємо інтерфейс VXLAN 5009 і знову генеруємо.

5. Захват керуючого трафіку в NSX

Ми можемо використовувати VirtualInfrastructureTrafficAnalysis для аналізу керуючого трафіку між хостом ESXi і контролером NSX і NSX менеджером. Трафік при спілкуванні зашифрований, але ми зможемо перевірити коректність заголовків на рівні 2, 3 або 4 моделі OSI.

На ESXi керуючий трафік відправляється й виходить через інтерфейс Vmkernel, тому для початку необхідно визначитися із точкою захвату.

Перший параметр, який необхідно вказати для VirtualInfrastructureTrafficAnalysis ця назва інтерфейсу Vmkernel –vmk. Інтерфейс керування на нашому ESXi називається vmk0. Далі нам необхідно вказати напрямок для захвату трафіку – точку захвату:

Таблиця 3.3 – Напрямок для захвату трафіку – точка захвату

Точка захвату	Використовується для
Portoutput	захвату пакетів від віртуального комутатора в інтерфейс Vmkernel
Portinput	захвату пакетів з інтерфейсу убік віртуального комутатора

На прикладі нижче ми захоплюємо трафік між хостом ESXi і контролером NSX. Керуючий трафік – це netcpa, який використовує порт 1234 у точці «5»:

Ми активуємо дві сесії для кожного з напрямків (хост ESXi – контролер NSX) у файл netcpad-to-controller.pcap.

Сесія для захвату netcpa у напрямку від контролера NSX убік хоста ESXi:

Надалі ми можемо об'єднати два різні файли в один за допомогою VirtualInfrastructureTrafficAnalysis для зручності аналізу двонаправленого трафіку, використовуючи функцію Merge.

3.3 Розробка функціональної схеми

Це не просто список самих корисних функцій – головний параметр корисності для нас був – користь/ціна.

Чим вище користь – тим вище функція у списку. Чим нижче ціна, тим вище функція в списку.

Так що Ви можете обмежитися тільки тими функціями, які, вважаєте, будуть корисні для Вас.

Розглянемо функції розробленого програмного забезпечення системи аналізу трафіку віртуальної інфраструктури:

1. Запуск декількох віртуальних серверів на одному фізичному

Це і є основне призначення віртуалізації.

Отут усі просто: чому купувати сервер для кожного застосунку, набагато більш вигідно додати в цей сервер більше пам'яті й використовувати могутніші процесори.

2. Робота з віртуальним сервером як зі звичайним файлом

Увесь віртуальний сервер – кілька файлів, які перебувають на дисках фізичного сервера або на системі зберігання даних.

Якщо Ви прагнете перенести віртуальний сервер – досить скопіювати 3-4 файлу, які перебувають у відповідній директорії.

Ці файли можна надалі скопіювати на інший фізичний сервер і весь віртуальний сервер буде працювати там.

3. Знімки віртуальних серверів

Знімки дозволяють фіксувати стан віртуального сервера й вертатися до цього стану в будь-який час.

Наприклад, у процесі налаштування якогось застосунку можна зробити знімок на початковий момент, після установки бази даних, до моменту налаштування.

Якщо щось пішло не так, можна перейти до будь-якого обраного знімка як минулому, що так і впливає.

Дуже сильно знімки допомагають саме в процесі установки нових сервісів.

Набагато простіше й швидше повернутися до минулого знімка, чим чистити реєстри і т.д. або заново встановлювати операційну систему.

4. Тонкі диски й просте керування ресурсами пам'яті й процесора віртуальних серверів

Будь-якому віртуальному серверу можна виділити стільки ресурсів, скільки Ви вважаєте потрібним. Наприклад, Ви підозрюєте, що через 2 року, віртуальному серверу може знадобиться 1Тб дискового простору.

Ви ставите йому прописуєте йому тонкий диск цього обсягу.

При цьому у Вас у розпорядженні сьогодні тільки диск на 600Гб.

Зрозуміло, через 2 року потрібно не забути купити ще дисків.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Ресурси пам'яті й процесора можна збільшувати й зменшувати протягом пари хвилин.

5. Централізоване керування всіма віртуальними серверами в інфраструктурі

На одному фізичному сервері залежно від його потужності може працювати від 5 до 100 віртуальних серверів (можливо й більше, усе буде залежати від фізичних можливостей сервера).

Уявимо собі ситуацію, що фізичних серверів 3, а віртуальних серверів, наприклад, 30-40.

Так це насправді велика кількість віртуальних серверів. І ними точно потрібно управляти централізовано.

6. Клонування віртуальних серверів і швидке розгортання із шаблону

Якщо Вам потрібно встановити новий додаток, Ви першою справою встановлюєте операційну систему для нього.

Шаблони дозволяють прискорити цей процес.

Єдине, що Вам знадобиться зробити – це прописати IP адреса новому віртуальному серверу.

Усередині цього шаблону може бути вже все, що потрібно. А шаблон являє собою ні що інше, як раніше налаштований Вами віртуальний сервер. Тобто спеціальних дій по налаштуванню, компонуванні й іншому для створення шаблону не треба.

7. Легке й надійне резервне копіювання віртуальних серверів

Гарна практика для віртуалізації – робити зовнішній блоковий бекап віртуальних серверів, а не як раніше, бекап зсередини операційної системи.

Ті самі файли, що вже згадувалися в п.2 складаються із блоків. Міняються у віртуальних серверах тільки ці блоки.

Перший раз засіб бекапу робить повну резервну копію, а потім додає блоки, що тільки змінилися (ну, залежно від політики, час від часу робить повні копії).

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47



Рисунок 3.2 – Функціональна схема системи

Ми рекомендуємо, якщо не потрібні функції, перераховані далі, використовувати VSphere Essentials (віртуалізація) і Veeam Backup & Replication (резервне копіювання).

8. Кластер високої доступності (деякі говорять відказостійкий кластер)

Ізюминка віртуалізації, без якої їсти шанс одержати проблеми в майбутньому.

Дійсно, якщо ми вирішили, що в нашому розпорядженні 30-40 віртуальних серверів, які працюють на 3-х фізичних серверах, тобто на кожному

фізичному більш 10 віртуальних, те що буде, якщо один з фізичних серверів зламається?

Кластер високої доступності допомагає відповісти на це питання так: через 2-3 хвилини віртуальні сервера, що працювали на цьому сервері запуснуть на 2-х фізичних серверах, що залишилися, і сервіси продовжать працювати. Налаштування такого кластера – усі що потрібно, щоб було саме так.

9. Жива міграція віртуальних серверів між фізичними серверами

Отут ми йдемо на поводу в громадськості й гарної назви, vmotion. Насправді технологія не така вже корисна. Єдине, коли вона дійсно допомагає – якщо потрібно виключити один з фізичних серверів.

Перед цим його віртуальні сервера можна «розтаскати» по інших серверах.

Ну й частково, щоб перерозподілити навантаження між фізичними серверами.

10. Реплікація віртуальних серверів, у тому числі філіальних через VPN або Інтернет

Майже як бекап, але в будь-який момент можна запустити віртуальний сервер – репліку, замість того, який перестав працювати. Якщо у філії або на вилученому майданчику був один сервер або більш (але був не налаштований кластер високої доступності), то це гарний вихід.

При цьому треба розуміти, що дані можуть пропасти, оскільки реплікація не безперервна.

Максимальний час, за які можуть бути загублені дані – 15 хвилин.

3.4 Розробка діаграми процесів

Відповідно до методичних рекомендацій розроблення графічної частини кваліфікаційної бакалаврської роботи розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Розроблена діаграма взаємодії процесів

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

використовується для представлення та візуалізації процесів обробки даних тобто структурного проектування бакалаврської роботи.

Основні складові елементи діаграми взаємодії процесів це потоки даних:

- Репозиторії, потік сховища даних.
- Потоки зовнішні по відношенню до системи сутності.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації 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.

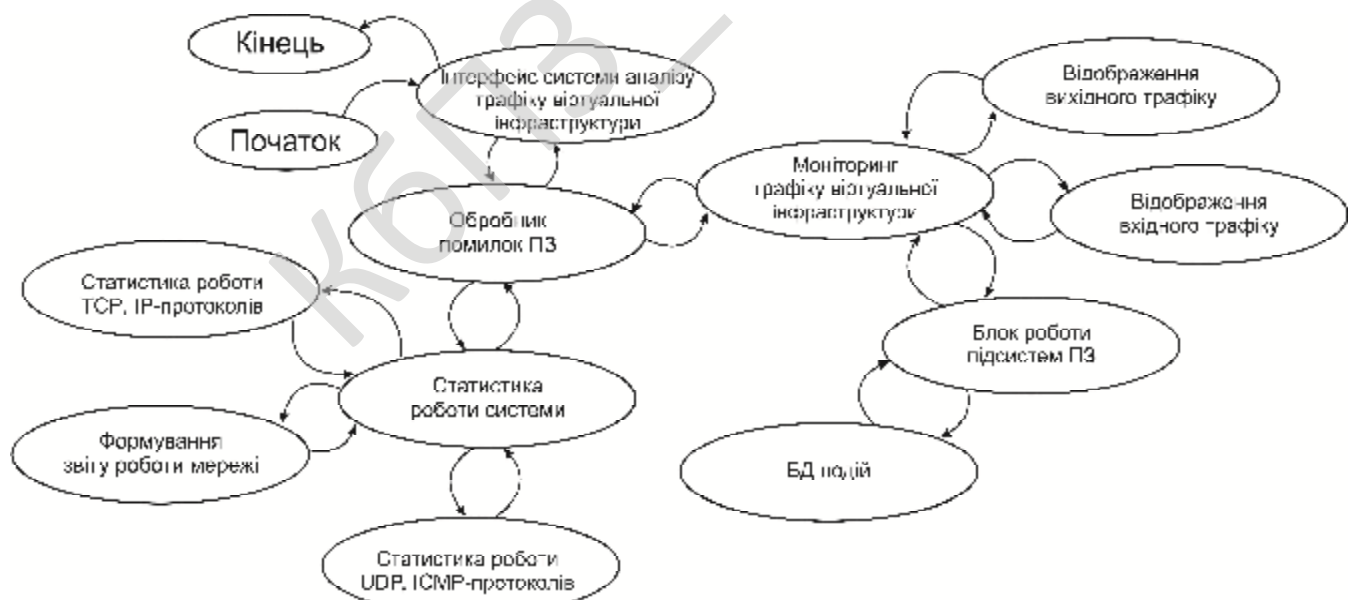


Рисунок 3.3 – Діаграма взаємодії процесів

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

К6ПЗ-2026

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

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.

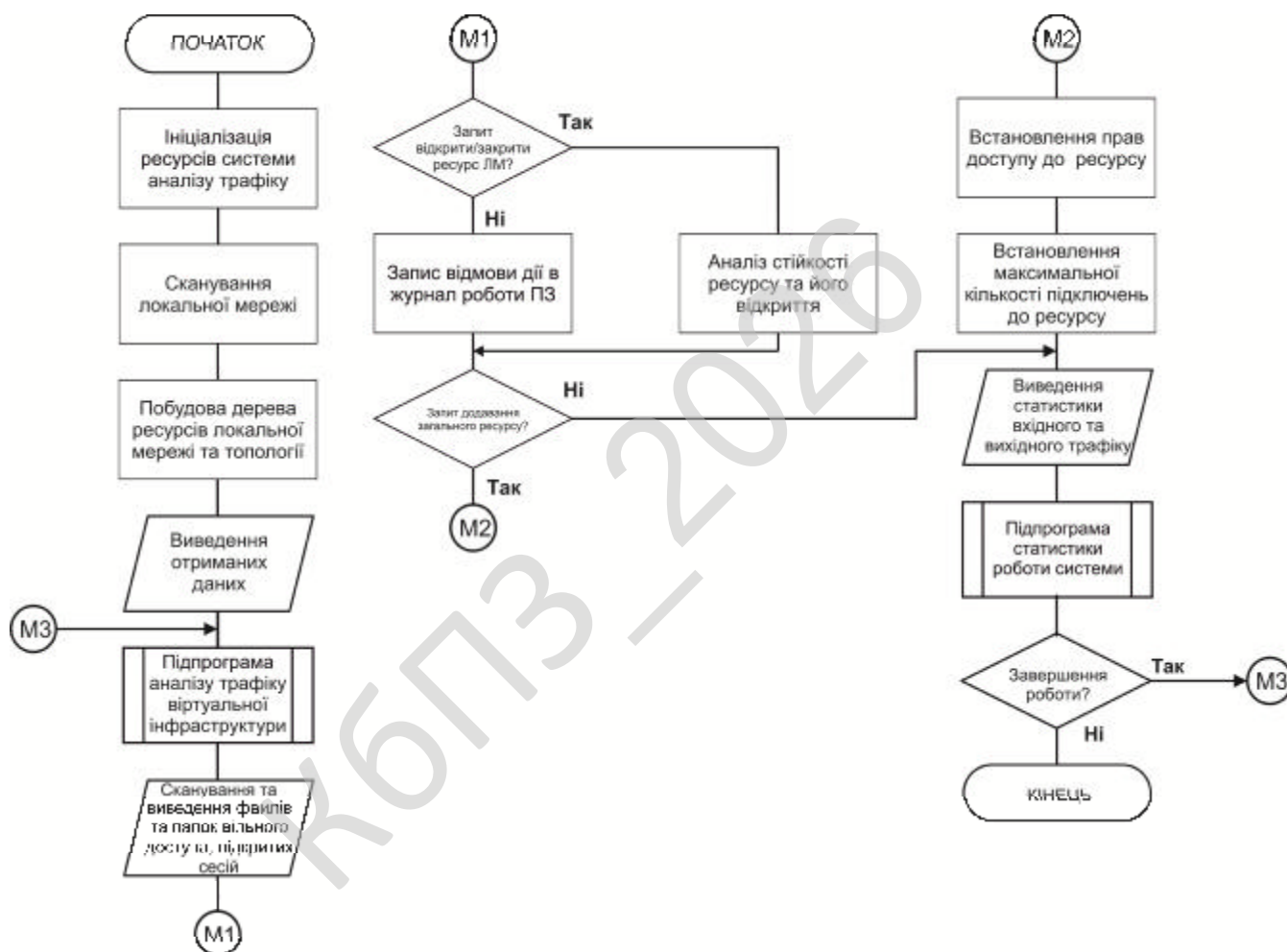
На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

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

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

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.



Розглянемо формат що використовується – JSON (JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) – це текстовий формат обміну даними між комп'ютерами.

JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією). Розробив і популяризував формат Дуглас Крокфорд.

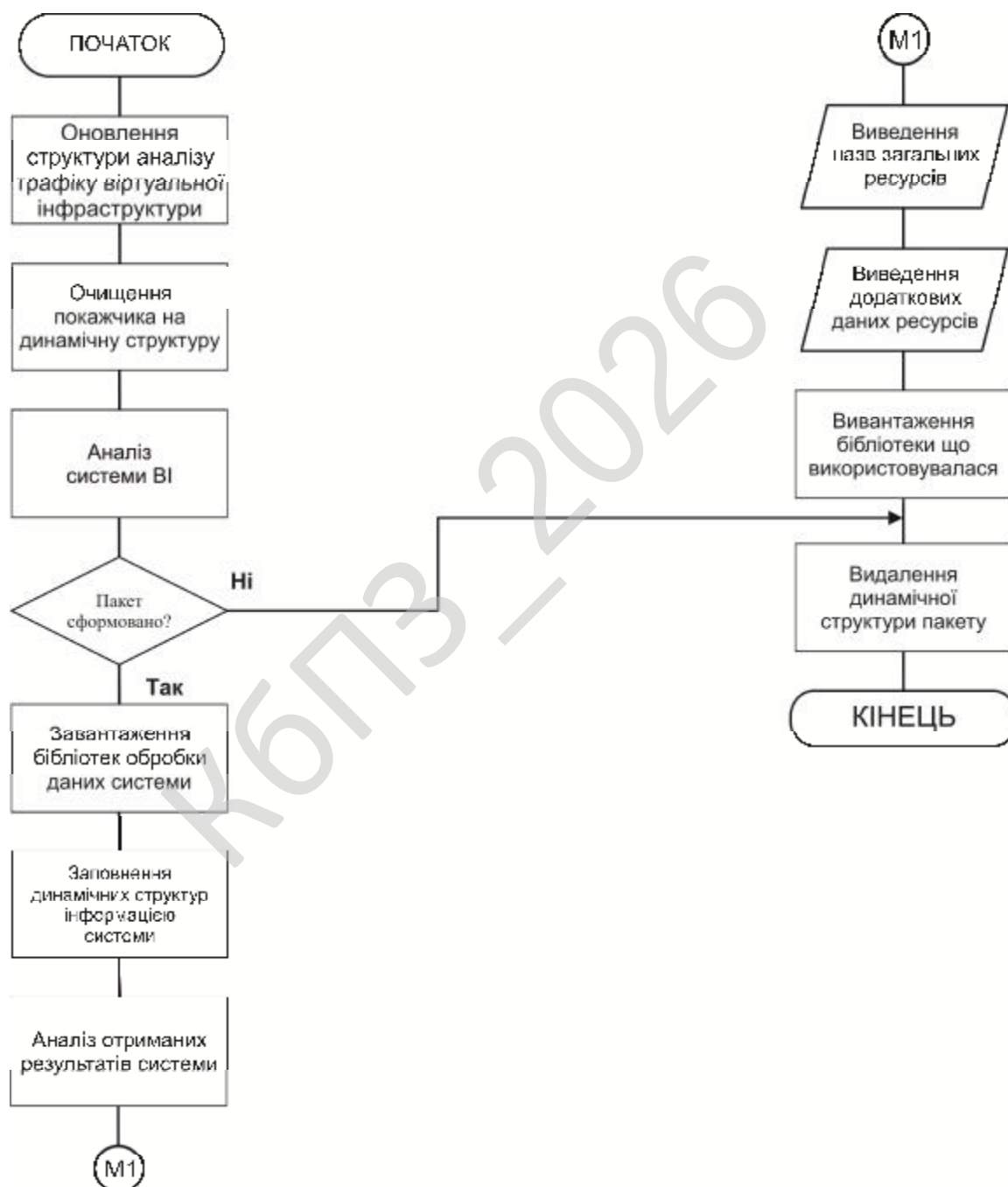


Рисунок 4.2 – Блок-схема роботи підпрограми

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

JSON з'явився через необхідність обміну даними з сервером у реальному часі без використання плагінів для браузерів, flash-додатків або Java-апплетів, які використовувались скрізь на початку 2000-х років. Дуглас Крокфорд був тим, хто активно просував новий на той час формат. Він з колегами хотів створити технологію, яка використовувала б можливості звичайного браузера давала б веб-розробникам можливість створювати веб-додатки із постійним двостороннім зв'язком із веб-сервером.

JSON вперше був використаний в проекті в Communities.com для Cartoon Network, він дозволяв обмінюватись повідомленнями і одночасно маніпулювати DHTML-елементами.

Веб-сайт JSON.org було запущено 2002 року. У грудні 2005-го року Yahoo! почав переводити деякі зі своїх веб-сервісів на роботу з JSON. Google взялася до роботи з технологією у своєму веб-протоколі GData у грудні 2006-го року.

Використання

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси).

Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

Приклад використання JSON (JavaScript)

```
const ajaxData = '{"name": "wiki", "fname": "pedia", "rates": [1, 4, 5, 6]}'
```

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

```
const ajaxObj = JSON.parse(ajaxData)
console.log(`${ajaxObj.name}_${ajaxObj.rates[2]}`)
```

Синтаксис

JSON будується на двох структурах:

1. Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.

2. Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

1. Об'єкт – це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом }. Кожне значення слідує за : і пари назва/значення відділяються комами.

2. Масив – це послідовність значень. Масив починається символом [і закінчується символом]. Значення відділяються комами.

3. Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

4. Рядок – це послідовність з нуля або більше символів юнікода, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком. Тип Рядок (String) дуже схожий на String в мовах C і Java. Число теж дуже схоже на C- або Java-число, за винятком того, що вісімкові та шістнадцяткові формати не використовуються. Пропуски можуть бути вставлені між будь-якими двома лексемами.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Наступний приклад показує JSON представлення об'єкта, що описує людину. У об'єкті є рядкові поля імені і прізвища, об'єкт, що описує адресу, і масив, що містить список телефонів.

```
{
  "firstName": "Іван",
  "address": {
    "postalCode": 21000
  },
  "phoneNumbers": [
    "044 123-1234",
    "050 123-4567"
  ]
}
```

Використання JSON в AJAX

Наступний фрагмент коду JavaScript показує, як клієнт може використати XMLHttpRequest для запиту об'єктів в форматі JSON з серверу. (Серверна частина коду пропущена, вона просто повертає на запит URL рядок у JSON форматі).

Треба відзначити, що тут використання XMLHttpRequest не є кросс-браузерним (за деталями звертайтеся на сторінку XMLHttpRequest), і код зазнаватиме незначних модифікацій в різних версіях оглядачів Internet Explorer, Opera, Safari або Mozilla. Використання запиту XMLHttpRequest обмежено правилом одного джерела (same origin policy): URL, що відповідає на запит, має посилатися на той же сайт, що обслуговує сторінку, що ініціювала запит.

Оглядачі можуть також використовувати тег <iframe> для асинхронного запиту JSON-даних в кросс-браузерному варіанті, або використати просте перенаправлення <form action="url_to_cgi_script" target="name_of_hidden_iframe">. Такий підхід був поширений до приходу популярного нині запиту XMLHttpRequest.

Динамічний тег <script> також можна використати для підвантаження JSON-даних. Ця техніка можлива, щоб обійти надто суворе правило одного

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

джерела, але вона не є безпечною. Запит JSONRequest пропонується, як безпечніша альтернатива.

Питання безпеки

Хоча JSON призначений для передачі даних в серіалізованому вигляді, його синтаксис відповідає синтаксису JavaScript і це створює низку проблем безпеки. Часто для обробки даних, отриманих від зовнішнього джерела у форматі JSON, до них застосовується функція `eval()` без якої-небудь попередньої перевірки.

JavaScript `eval()`

Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції `eval()`, яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових парсерів.

Техніка використання `eval()` робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку автентифікації. Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання `eval()` отримані від зовнішнього джерела дані можуть перевірятися за допомогою регулярних виразів. У RFC, що визначає JSON пропонується використовувати такий код для перевірки його відповідності формату JSON

```
const my_JSON_object = !(/[^\s:{}\[\]\0-9.\-+Eaeflnr-u
\n\r\t]/.test(
  text.replace(/"(\.|\["\"])*"/g, ''))) &&
eval('(' + testedData + ')');
```

Як безпечніша альтернатива `eval()` була запропонована нова функція `parseJSON()`, здатна обробляти тільки JSON-дані.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Вбудований JSON

Останні версії веб-браузерів мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери: Mozilla Firefox 3.5+; SeaMonkey 2; Thunderbird 3; Microsoft Internet Explorer 8; Opera 10.5; Браузери, засновані на WebKit (наприклад, Google Chrome, Apple Safari).

Принаймні дві популярні бібліотеки JavaScript використовують вбудований JSON у разі його доступності: jQuery; Dojo.

Підробка крос-доменного запиту

Непродумане використання JSON робить сайти вразливими до підробки міжсайтових запитів (CSRF або XSRF). Оскільки тег <script> допускає використання джерела, що не належить до того ж домену, що і використовуваний ресурс, це дозволяє виконувати код даних, представлених у форматі JSON, в контексті довільної сторінки, що робить можливою компрометацію паролів або іншої конфіденційної інформації користувачів, що пройшли авторизацію на іншому сайті.

Це є проблемою тільки у разі вмісту в JSON-даних конфіденційної інформації, яка може бути компрометована третьою стороною і якщо сервер розраховує на політику одного джерела, блокуючи доступ до даних при виявленні зовнішнього запиту. Це не є проблемою, якщо сервер визначає допустимість запиту, надаючи дані тільки у разі його коректності. HTTP cookie не можна використовувати для визначення цього. Виключне використання HTTP cookie використовується підробкою міжсайтових запитів.

Розширення, JSONP

JSONP або «JSON з підкладкою» є розширенням JSON, коли назва функції зворотного виклику вказується як вхідний аргумент. Спочатку ідея була запропонована в блозі MacPython в 2005 році, і в наш час використовується

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

багатьма Web 2.0 застосунками, такими, як Dojo Toolkit Applications, Google Toolkit Applications і zanox Web Services.

Подальші розширення цього протоколу були запропоновані з урахуванням введення додаткових аргументів, як, наприклад, у разі JSONPP за підтримки S3DB вебсервісів.

Оскільки JSONP використовує скрипт-теги, виклики по суті відкриті світові. З цієї причини JSONP може бути недоречними для зберігання конфіденційних даних.

Включення скриптових тегів від віддалених сайтів дозволяє їм передати будь-який контент на сайті. Якщо віддалений сайт має вразливості, які дозволяють виконати ін'єкції JavaScript, то початковий сайт також може бути ними зачеплений.

Розширення, BSON

BSON – це бінарна форма представлення простих структур даних і асоціативних масивів (які називають об'єктами або документами). Назва «BSON» заснована на визначенні JSON і неофіційно означає «Binary JSON» (бінарний JSON).

JSON Reference

Стандарт JSON не описує посилання на інші об'єкти або частини, але існує чернетка стандарту IETF для посилань на об'єкти на основі JSON. Посилання дозволяють здійснювати трансклюзію - вставляти одні документи в інші.

JSON Reference - це JSON-об'єкт з ключем "\$ref" (всі інші ключі ігноруються) і значенням стрічкового типу що містить URI, наприклад:

```
{ "$ref": "http://example.com/example.json#/foo/bar" }
```

Якщо URI містить ідентифікатор фрагмента (в прикладі вище "/foo/bar"), він інтерпретується як JSON Pointer.

Модуль doxox.json.ref в Dojo toolkit, забезпечує підтримку декількох форм JSON Reference.

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

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Гнучка́ розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між самоорганізовуваними багатофункціональними командами.

Гнучка розробка – найкращий засіб для підвищення продуктивності розробників програмного забезпечення.

Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі, і включає всі завдання, необхідні для видачі мінімального приросту за функціональністю: планування, аналіз вимог, проектування, кодування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі те, що гнучкий програмний проект готовий до випуску наприкінці кожної ітерації. Після закінчення кожної ітерації, команда виконує переоцінку пріоритетів розробки.

Agile акцентує увагу на безпосередньому спілкуванні «віч-на-віч». Більшість agile команд розташовані в одному офісі, його іноді називають bullpen. Як мінімум вона включає і «замовників» (замовники, які визначають продукт, також це можуть бути менеджери продукту, бізнес аналітики або клієнти). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних авторів і менеджерів.

Основною метрикою agile методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це привело до критики цих методів як недисциплінованих.

Agile – родина процесів розробки, а не єдиний підхід в розробці програмного забезпечення, і визначається Agile Manifesto. Agile не включає практик, а визначає цінності та принципи, якими керуються успішні команди.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Agile Manifesto розроблений і прийнятий 17 розробниками 11-13 лютого 2001 року на лижному курорті The Lodge at Snowbird в горах Юти. Маніфест підписали представники наступних методологій Extreme programming, Scrum, DSDM, Adaptive software development, Crystal Clear, Feature driven development, Pragmatic Programming. Agile Manifesto містить 4 основні ідеї та 12 принципів. Примітно, що Agile Manifesto не містить практичних порад.

Основні ідеї:

- Особистості та їхні взаємодії важливіші, ніж процеси та інструменти;
- Робоче програмне забезпечення важливіше, ніж повна документація;
- Співпраця із замовником важливіша, ніж контрактні зобов'язання;
- Реакція на зміни важливіша, ніж дотримання плану.

Принципи, які роз'яснює Agile Manifesto:

- задоволення клієнта за рахунок ранньої та безперебійної поставки коштовного програмного забезпечення;
- вітання змін вимог навіть наприкінці розробки (це може підвищити конкурентоспроможність отриманого продукту);
- часта поставка робочого програмного забезпечення (кожен місяць або тиждень або ще частіше);
- тісне, щоденне спілкування замовника з розробниками впродовж всього проекту;
- проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;
- рекомендований метод передачі інформації – особиста розмова (віч-на-віч);
- робоче програмне забезпечення – найкращий вимірювач прогресу;
- спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійну увагу поліпшенню технічної майстерності та зручному дизайну;
- простота – мистецтво не робити зайвої роботи;

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

– найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованої команди;

– постійна адаптація до мінливих обставин.

Маніфест та Принципи гнучкої розробки містять високорівневі ідеї щодо того, як потрібно вибудовувати процес розробки програмного забезпечення, щоб успішно завершувати проекти й створювати команди, в яких приємно та цікаво працювати.

Документи визначають, що потрібно для цього зробити, але не говорять, як це зробити. По-іншому й не могло бути, оскільки Маніфест та Принципи народилися внаслідок консенсусу представників різних (хоча й споріднених) напрямів, які могли знайти спільну основу лише на рівні базових цінностей та принципів.

Критика. Багато керівників проектів, що працюють у традиційних методологіях на кшталт «водоспаду», критикують agile-методи.

Один з повторюваних пунктів критики: при agile-підході часто нехтують створенням «дорожньої карти» розвитку продукту, так само як і управлінням вимогами, в процесі якого і формується така «карта».

Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість замовника раптом і несподівано наприкінці кожної ітерації виставляти нові вимоги, що часто суперечать архітектурі вже створеного і поставленого продукту.

Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації.

Крім того вважається, що робота в agile мотивує розробників вирішувати всі прибулі завдання найпростішим і найшвидшим можливим способом, при цьому часто не звертаючи уваги на коректність коду з точки зору вимог базової платформи (підхід «працює, та й добре», при цьому не враховується, що може

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

перестати працювати при найменшій зміні або ж породити важкі до відтворення дефекти після реального розгортання у клієнта).

Це призводить до зниження якості продукту і накопиченню дефектів.

Методології. Існують методології, які дотримуються цінностей і принципів заявлених в Agile Manifesto, деякі з них:

1. Agile Modeling – набір понять, принципів і прийомів (практик), що дозволяють швидко і просто виконувати моделювання і документування в проектах розробки програмного забезпечення. Не включає в себе детальну інструкцію з проектування, не містить описів, як будувати діаграми на UML.

Основна мета – ефективне моделювання і документування; але не охоплює програмування та тестування, не включає питання управління проектом, розгортання і супроводу системи. Однак включає в себе перевірку моделі кодом.

2. Agile Unified Process (AUP) спрощена версія IBM Rational Unified Process (RUP), розроблена Скоттом Амблером, яка описує просте і зрозуміле наближення (модель) для створення програмного забезпечення для бізнес-додатків.

3 Agile Data Method – група ітеративних методів розробки програмного забезпечення, в яких вимоги та рішення досягаються в рамках співпраці різних крос-функціональних команд.

4. DSDM заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD). Являє собою ітеративний і інкрементний підхід, який надає особливого значення тривалій участі в процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка. Використовуване в FDD поняття функції або властивості (feature) системи досить близько до поняття прецеденту використання, використовуваному в RUP, істотна

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

відмінність – це додаткове обмеження: «кожна функція повинна допускати реалізацію не більше, ніж за два тижні».

Тобто якщо сценарій використання досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

8. Getting Real – ітераційний підхід без функціональних специфікацій, що використовується для веб-додатків. У даному методі спершу розробляється інтерфейс програми, а потім її функціональна частина.

9. OpenUP – це ітераційно-інкрементний метод розробки програмного забезпечення. Позиціюється, як легкий і гнучкий варіант RUP. OpenUP ділить життєвий цикл проекту на чотири фази: початкова фаза, фази уточнення, конструювання та передачі. Життєвий цикл проекту забезпечує надання зацікавленим особам та членам колективу точок ознайомлення і прийняття рішень впродовж усього проекту.

Це дозволяє ефективно контролювати ситуацію і вчасно приймати рішення про задовільність результатів. План проекту визначає життєвий цикл, а кінцевим результатом є остаточний додаток.

10. Scrum встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги або вносячи тактичні зміни. Використання цієї методології дає можливість виявляти і усувати відхилення від бажаного результату на більш ранніх етапах розробки програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Використовує підходи з концепції бережливого виробництва.

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок.

До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

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

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

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

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

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

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

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

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

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

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан.

У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

– Виправлено (виправлення включені у версію таку-то).

- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).

– «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок.

Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм REDOC III, який оперує з 80-бітовим блоком.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Довжина ключа може змінюватися й досягати 2560 байт (204800 біт). Алгоритм складається тільки з операцій XOR над байтами ключа й відкритого тексту, перестановки й підстановки не використовуються.

1. Створюють таблицю ключів з 256 10-байтових ключів, використовуючи секретний ключ.

2. Створюють два 10-байтових блоки масок M1 і M2. M1 являє собою результат операції XOR перших 128 10-байтових ключів, а M2 – результат операції XOR других 128 10-байтових ключів.

3. Для шифрування 10-байтового блоку:

а. Виконують операцію XOR з першим байтом блоку даних і першим байтом M1. Вибирають ключ у таблиці ключів, розрахованої в раунді 1. Використовують обчислене значення XOR як індекс таблиці. Виконують операцію XOR з кожним, крім першого, байтом блоку даних і відповідним байтом обраного ключа.

б. Виконують операцію XOR із другим байтом блоку даних і другим байтом M1. Вибирають ключ у таблиці ключів, розрахованої в раунді 1. Використовують обчислене значення XOR як індекс таблиці. Виконаєте операцію XOR з кожним, крім другого, байтом блоку даних і відповідним байтом обраного ключа.

с. Продовжують ці дії з усім блоком даних (з 3-10 байтами), поки не буде використаний кожний байт для вибору ключа з таблиці після виконання операції XOR з ним і відповідним значенням M1. Потім виконують операцію XOR з кожним, крім використаного для вибору ключа, байтом, і ключем.

д. Повторюють етапи а-с для M2.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1. З рисунку головного вікна можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Функціональних кнопок ПЗ: Закрити/Відкрити; Оновити; Налаштувати; Додати; Моніторинг; Статистика.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші: Налаштування.
- Верхнього меню: Система; Налаштування; Довідка.
- Розділу виведення результату роботи системи.

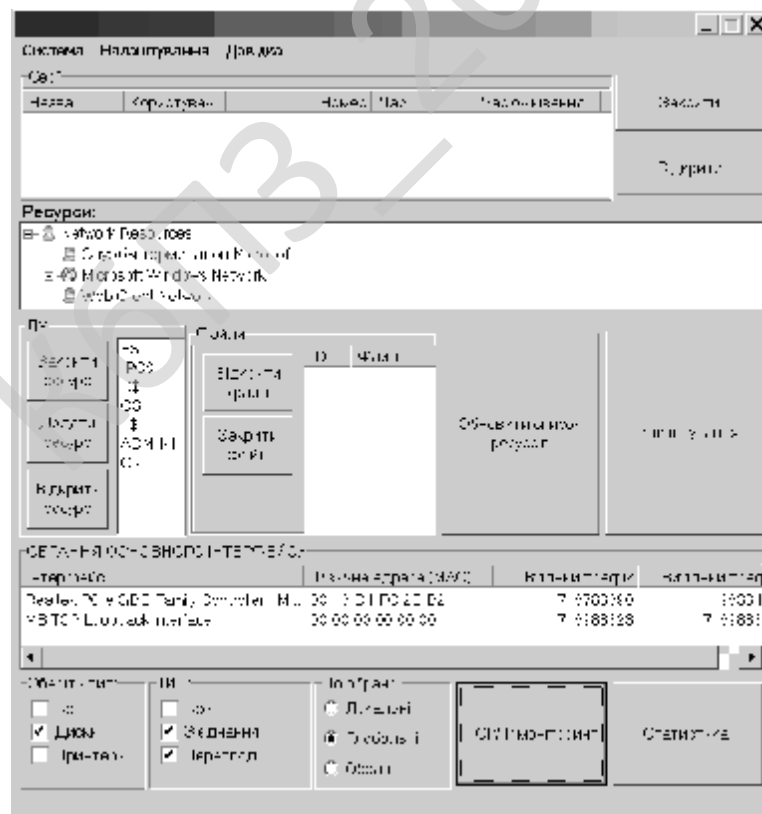


Рисунок 5.1 – Головне вікно ПЗ

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. 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.

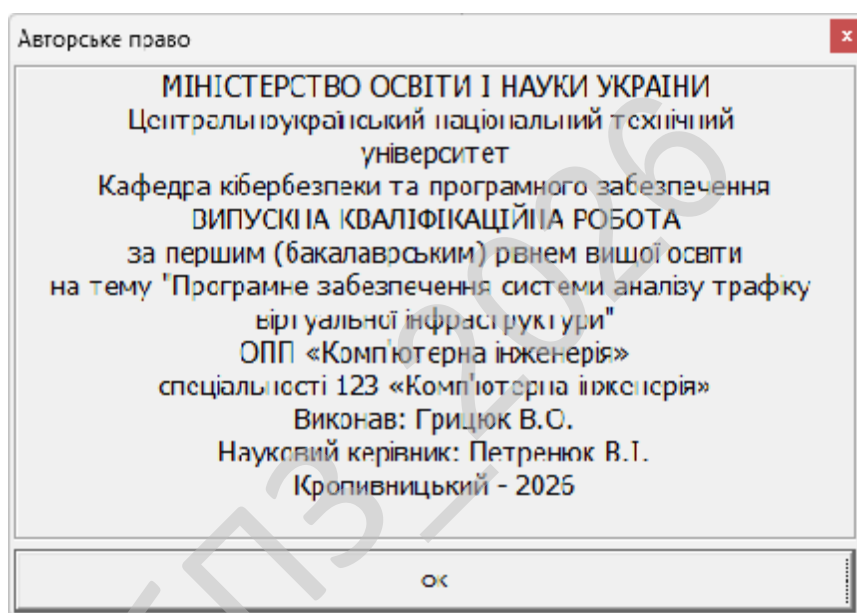


Рисунок 5.2 – Авторське право

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.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таким чином у результаті вищерозглянутого можна стверджувати що розроблено інтерфейс системи у відповідності з вибраною метою роботи.

Система містить максимальний необхідний набір функцій придатних для виконання будь-яких дій для забезпечення повноцінної роботи програми.

Далі розглянемо висновки та використані літературні джерела.

К6ПЗ_2026

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи аналізу трафіку віртуальної інфраструктури.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем аналізу трафіку віртуальної інфраструктури.
- Досліджена система аналізу трафіку віртуальної інфраструктури.
- На основі отриманих результатів досліджень створена програмна реалізація системи аналізу трафіку віртуальної інфраструктури.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання аналізу трафіку віртуальної інфраструктури.

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

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи аналізу трафіку віртуальної інфраструктури. 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

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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 REDOC III.

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

К6ПЗ-2026

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
2. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
3. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
4. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
5. 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.
6. 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.
7. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.
8. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

9. 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.

10. 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.

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

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

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

21. 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.

22. 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.

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

23. 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.

24. 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.

25. 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.

26. 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.

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

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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

38. 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.

39. 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.

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

41. 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.

42. Смірнов О.А., Дресва Г.М., Дресв 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.

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

44. 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).

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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

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

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

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

49. 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.

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

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

52. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережевих експертів безпечної маршрутизації у хмарних

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

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

54. 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.

55. 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.

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

					ВКРБ-123.26.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83