Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Програмні засоби моніторингу інформаційної системи

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-1
Назар КОЛОБАЄВ
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник: ас. Ірина КЛИМОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Колобаєву Назару Михайловичу
(прізвище, ім’я, по батькові)

1. Тема роботи Програмні засоби моніторингу інформаційної системи

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи
моніторинг
інформаційна система
критерії надійності

4. Перелік питань, що потрібно опрацювати у роботі
Теоретичні основи моніторингу інформаційних систем
Аналіз сучасних інструментів та підходів до моніторингу
Розробка та реалізація програмного засобу моніторингу
Оцінка ефективності та перспективи розвитку

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 13 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих рішень та алгоритмів	31.05.2025–03.06.2025	
3	Вибір архітектури системи	04.06.2025–06.06.2025	
4	Вибір програмних засобів	07.06.2025–08.06.2025	
5	Програмна реалізація	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання " 26 " травня 2025 р.

Здобувач  (підпис)

Керівник роботи _____ ас. Ірина КЛИМОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 64 с., 12 рис., 2 дод., 9 джерел.

МОНІТОРИНГ ІНФОРМАЦІЙНИХ СИСТЕМ, ПРОГРАМНІ ЗАСОБИ МОНІТОРИНГУ, ІТ-ІНФРАСТРУКТУРА, НАДІЙНІСТЬ СИСТЕМ, ПРОДУКТИВНІСТЬ, МАСШТАБОВАНІСТЬ, АРХІТЕКТУРА СИСТЕМ МОНІТОРИНГУ, ЗБІР ДАНИХ, ОБРОБКА ДАНИХ, ВІЗУАЛІЗАЦІЯ ДАНИХ, ВИЯВЛЕННЯ АНОМАЛІЙ, ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, МІКРОСЕРВІСНА АРХІТЕКТУРА, ІНТЕГРАЦІЯ СИСТЕМ, БЕЗПЕКА ДАНИХ, АВТОМАТИЗАЦІЯ, ПРЕДИКТИВНА АНАЛІТИКА, ZABBIX, NAGIOS, PROMETHEUS, GRAFANA.

Метою кваліфікаційної роботи є аналіз сучасних підходів до моніторингу стану інформаційних систем, а також розробка програмних засобів моніторингу стану інформаційної системи.

У роботі досліджено сучасні підходи до моніторингу стану інформаційних систем та проаналізовано програмні засоби, що забезпечують ефективний контроль функціонування ІТ-інфраструктури. Проведено комплексний аналіз теоретичних основ моніторингу інформаційних систем, включаючи дослідження концептуальних засад, критеріїв надійності та доступності систем, а також класифікацію існуючих рішень.

Виконано порівняльний аналіз провідних програмних засобів моніторингу, таких як Zabbix, Nagios, Prometheus та Grafana, з детальним вивченням їх архітектурних особливостей, функціональних можливостей та принципів роботи. Особливу увагу приділено проблемам масштабування, продуктивності та забезпечення безпеки в контексті сучасних вимог до систем моніторингу.

ABSTRACT

Bachelor’s thesis: 64 pages, 12 figures, 2 appendices, 9 sources.

INFORMATION SYSTEMS MONITORING, MONITORING SOFTWARE TOOLS, IT INFRASTRUCTURE, SYSTEM RELIABILITY, PERFORMANCE, SCALABILITY, MONITORING SYSTEM ARCHITECTURE, DATA COLLECTION, DATA PROCESSING, DATA VISUALIZATION, ANOMALY DETECTION, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, MICROSERVICE ARCHITECTURE, SYSTEM INTEGRATION, DATA SECURITY, AUTOMATION, PREDICTIVE ANALYTICS, ZABBIX, NAGIOS, PROMETHEUS, GRAFANA.

The major goal of this thesis is is to analyze modern approaches to information systems monitoring and to develop software tools for monitoring the state of information systems.

In order to a A comparative analysis of leading monitoring software tools such as Zabbix, Nagios, Prometheus, and Grafana has been performed, with detailed examination of their architectural features, functional capabilities, and operating principles. Special attention has been given to scalability, performance, and security issues in the context of modern requirements for monitoring systems.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – програмний інтерфейс застосунків

AWS – Amazon Web Services

BMC – BMC Software

CRM – система управління взаємовідносинами з клієнтами

ERP – система планування ресурсів підприємства

HTTP – протокол передачі гіпертексту

IPMI – інтелектуальний інтерфейс управління платформою

IT – інформаційні технології

ITSM – управління IT-сервісами

JMX – розширення управління Java

PRTG – мережевий монітор PRTG

REST – архітектурний стиль передачі стану репрезентації

SIEM – управління інформацією та подіями безпеки

SNMP – простий протокол управління мережею

SSL – рівень безпечних з'єднань

TLS – безпека транспортного рівня

WMI – інструментарій управління Windows

ВСТУП

Сучасний етап розвитку інформаційних технологій характеризується експоненційним зростанням складності інформаційних систем та їх критичної важливості для функціонування організацій різних сфер діяльності. В умовах цифрової трансформації суспільства інформаційні системи стають основою для прийняття стратегічних рішень, забезпечення безперервності бізнес-процесів та підтримки конкурентоспроможності підприємств. Водночас, зростання масштабів та складності інформаційних систем супроводжується підвищенням ризиків виникнення збоїв, відмов та порушень їх функціонування.

Забезпечення стабільної роботи інформаційних систем вимагає впровадження комплексних підходів до моніторингу їх стану, що дозволяє своєчасно виявляти потенційні проблеми, попереджувати критичні ситуації та мінімізувати час простою системи. Традиційні методи контролю стану інформаційних систем, засновані на реактивному підході, виявляються недостатніми для забезпечення необхідного рівня надійності та доступності сучасних IT-інфраструктур.

Актуальність розробки програмних засобів моніторингу стану інформаційних систем обумовлена необхідністю переходу від реактивного до проактивного управління IT-інфраструктурою, що дозволяє значно підвищити ефективність експлуатації інформаційних систем та зменшити витрати на їх обслуговування. Розвиток технологій штучного інтелекту, машинного навчання та аналітики великих даних відкриває нові можливості для створення інтелектуальних систем моніторингу, здатних не лише виявляти проблеми, але й прогнозувати їх виникнення.

Метою роботи є аналіз сучасних підходів до моніторингу стану інформаційних систем, а також розробка програмних засобів моніторингу стану інформаційної системи.

Досягнення поставленої мети передбачає вирішення наступних завдань. Необхідно провести теоретичний аналіз концептуальних основ моніторингу інформаційних систем, дослідити еволюцію підходів до забезпечення надійності та доступності IT-інфраструктури. Важливим завданням є систематизація та класифікація існуючих програмних рішень для моніторингу, виявлення їх переваг та обмежень. Дослідження повинно включати порівняльний аналіз архітектурних рішень та функціональних можливостей провідних систем моніторингу. Необхідно також дослідити проблеми масштабування, продуктивності та безпеки в контексті сучасних вимог до систем моніторингу.

Практична частина дослідження передбачає розробку програмного засобу моніторингу, що враховує сучасні тенденції розвитку IT-технологій. Важливим завданням є оцінка ефективності запропонованих рішень та формулювання рекомендацій щодо їх практичного застосування в різних сферах діяльності.

1 ТЕОРЕТИЧНІ ОСНОВИ МОНІТОРИНГУ ІНФОРМАЦІЙНИХ СИСТЕМ

1.1 Поняття інформаційної системи та її життєвий цикл

Інформаційна система являє собою організовану сукупність апаратних, програмних, мережевих та людських ресурсів, призначених для збору, зберігання, обробки та поширення інформації з метою підтримки прийняття рішень та управління організацією. Сучасні інформаційні системи характеризуються високим рівнем складності, гетерогенністю компонентів та критичною важливістю для функціонування організацій.

Рисунок 1.1 – Життєвий цикл ІС

Архітектура сучасних інформаційних систем базується на принципах модульності, масштабованості та відкритості. Типова інформаційна система включає рівень представлення даних, рівень бізнес-логіки, рівень доступу до даних та рівень зберігання даних. Кожен з цих рівнів може бути реалізований з використанням різних технологій та платформ, що створює додаткові

виклики для забезпечення цілісності та надійності системи.

Життєвий цикл інформаційної системи (рисунок 1.1) охоплює етапи планування, аналізу, проектування, реалізації, тестування, впровадження, експлуатації та виведення з експлуатації. Кожен етап характеризується специфічними вимогами до моніторингу та контролю стану системи. На етапі планування важливо визначити критерії ефективності та показники якості, які будуть використовуватися для оцінки функціонування системи. Етап аналізу передбачає детальне дослідження вимог до моніторингу та визначення архітектурних рішень.

Проектування системи моніторингу повинно здійснюватися паралельно з проектуванням самої інформаційної системи, що дозволяє забезпечити оптимальну інтеграцію компонентів моніторингу. Етап реалізації включає розробку програмних компонентів моніторингу та їх інтеграцію з основною системою. Тестування системи моніторингу повинно проводитися як окремо, так і в складі загальної системи.

Етап експлуатації є найбільш тривалим та важливим з точки зору моніторингу, оскільки саме на цьому етапі система моніторингу повинна забезпечувати безперервний контроль стану інформаційної системи. Важливим аспектом є адаптація системи моніторингу до змін в інформаційній системі, що відбуваються в процесі її еволюції та модернізації.

1.2 Критерії надійності та доступності інформаційної системи

Надійність інформаційної системи (рисунок 1.2) характеризується її здатністю виконувати задані функції в заданих умовах експлуатації протягом заданого проміжку часу. Це комплексний показник, що включає такі аспекти як безвідмовність, довговічність, ремонтопридатність та збережуваність. Безвідмовність характеризує властивість системи безперервно зберігати працездатність протягом деякого інтервалу часу. Довговічність відображає

властивість системи зберігати працездатність до настання граничного стану при встановленій системі технічного обслуговування та ремонту.

Ремонтопридатність характеризує пристосованість системи до попередження та виявлення причин виникнення відмов, пошкоджень та усунення їх наслідків шляхом проведення ремонтів та технічного обслуговування. Збережуваність відображає властивість системи зберігати показники безвідмовності, довговічності та ремонтопридатності протягом та після зберігання та транспортування.

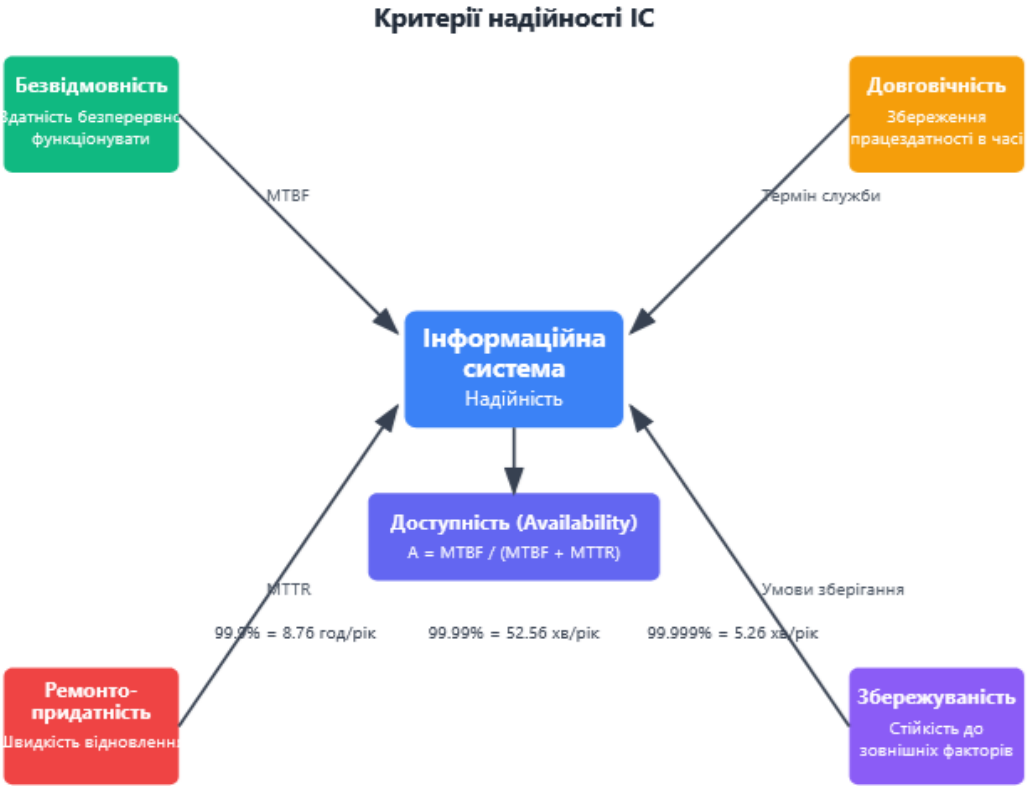

Рисунок 1.1 – Критерії надійності ІС

Доступність інформаційної системи визначається як ймовірність того, що система буде функціонувати належним чином у довільний момент часу. Цей показник тісно пов'язаний з надійністю системи, але має більш практичну спрямованість, оскільки характеризує фактичну здатність користувачів отримувати доступ до послуг системи. Доступність зазвичай виражається у відсотках або у вигляді коефіцієнта готовності.

Для кількісної оцінки надійності та доступності використовуються

різні метрики. Середній час між відмовами характеризує середній інтервал часу між послідовними відмовами системи. Середній час відновлення визначає середній час, необхідний для відновлення працездатності системи після виникнення відмови. Коефіцієнт готовності розраховується як відношення середнього часу між відмовами до суми середнього часу між відмовами та середнього часу відновлення.

Забезпечення високого рівня надійності та доступності вимагає комплексного підходу, що включає використання надійних компонентів, впровадження механізмів резервування та відмовостійкості, організацію ефективного технічного обслуговування та моніторингу. Система моніторингу відіграє ключову роль у забезпеченні надійності та доступності, оскільки дозволяє своєчасно виявляти потенційні проблеми та попереджувати критичні ситуації.

1.3 Поняття моніторингу ІС: цілі, завдання, типи

Моніторинг інформаційних систем являє собою безперервний процес спостереження, збору, аналізу та інтерпретації даних про стан та функціонування різних компонентів системи з метою забезпечення її оптимальної роботи. Це комплексна діяльність, що охоплює як технічні аспекти контролю стану обладнання та програмного забезпечення, так і аналіз бізнес-процесів та задоволеності користувачів.

Основною метою моніторингу є забезпечення безперервної та ефективної роботи інформаційної системи шляхом проактивного виявлення та усунення проблем до їх критичного впливу на функціонування системи. Це передбачає не лише реактивне реагування на виникаючі проблеми, але й прогнозування потенційних збоїв та попереджувальні заходи.

Завдання моніторингу включають безперервне спостереження за станом апаратних компонентів системи, включаючи сервери, мережеве обладнання, системи зберігання даних та інші елементи IT-інфраструктури.

Важливим завданням є контроль за функціонуванням програмного забезпечення, включаючи операційні системи, системи управління базами даних, прикладні програми та сервіси.

Рисунок 1.3 – Типи та рівні моніторингу ІС

Моніторинг (рисунок 1.3) повинен забезпечувати контроль за продуктивністю системи, включаючи час відгуку, пропускну здатність, використання ресурсів та інші ключові показники ефективності. Необхідним є також моніторинг безпеки системи, включаючи виявлення спроб несанкціонованого доступу, аналіз журналів безпеки та контроль за дотриманням політик безпеки.

За способом реалізації розрізняють пасивний та активний моніторинг. Пасивний моніторинг базується на аналізі даних, що генеруються самою системою в процесі її функціонування, включаючи журнали подій, системні повідомлення та статистичну інформацію. Активний моніторинг передбачає цілеспрямоване тестування системи шляхом генерації контрольних запитів та аналізу відповідей.

За рівнем деталізації розрізняють моніторинг на рівні інфраструктури, який зосереджується на контролі стану апаратних компонентів та базового програмного забезпечення, моніторинг на рівні застосунків, що фокусується на функціонуванні прикладних програм та сервісів, та моніторинг на рівні бізнес-процесів, який аналізує ефективність виконання бізнес-функцій.

1.4 Існуючі програмні рішення для моніторингу: огляд і класифікація

Сучасний ринок програмних засобів моніторингу інформаційних систем характеризується великою різноманітністю рішень, що відрізняються за функціональними можливостями, архітектурними особливостями, цільовими сферами застосування та ліцензійними умовами. Аналіз існуючих рішень дозволяє виділити декілька основних категорій програмних засобів моніторингу (рисунок 1.4).

Універсальні платформи моніторингу являють собою комплексні рішення, здатні контролювати різні аспекти функціонування інформаційних систем. До цієї категорії належать такі відомі продукти як Zabbix, який забезпечує моніторинг мережевої інфраструктури, серверів, застосунків та сервісів. Nagios відзначається гнучкістю налаштування та широкими можливостями розширення через систему плагінів. SolarWinds пропонує комплексне рішення для моніторингу корпоративних IT-інфраструктур.

Спеціалізовані рішення для моніторингу застосунків зосереджуються на детальному аналізі продуктивності та доступності прикладних програм. New Relic та Dynatrace пропонують глибокий аналіз продуктивності застосунків з використанням технологій штучного інтелекту. AppDynamics забезпечує моніторинг повного ланцюга виконання бізнес-транзакцій.

Рішення для моніторингу інфраструктури фокусуються на контролі стану апаратних компонентів та базового програмного забезпечення. PRTG Network Monitor спеціалізується на моніторингу мережевої інфраструктури. ManageEngine OpManager забезпечує комплексний моніторинг IT-

інфраструктури підприємства.

Рисунок 1.4 – Порівняльний аналіз існуючих рішень

Opensource рішення представляють альтернативу комерційним продуктам та відзначаються гнучкістю налаштування та можливістю адаптації під специфічні потреби організації. Prometheus разом з Grafana формують потужну платформу для збору метрик та їх візуалізації. Icinga являє собою розвиток проекту Nagios з поліпшеною архітектурою та функціональністю.

Хмарні рішення для моніторингу набувають все більшої популярності завдяки простоті впровадження та масштабованості. Amazon CloudWatch забезпечує моніторинг ресурсів AWS. Google Cloud Monitoring пропонує аналогічні можливості для Google Cloud Platform. Datadog являє собою універсальну хмарну платформу моніторингу.

Класифікація за архітектурними особливостями дозволяє виділити централізовані системи, де всі компоненти моніторингу керуються з єдиного центру, та розподілені системи, що забезпечують більшу гнучкість та масштабованість. Агентні системи вимагають встановлення спеціального програмного забезпечення на моніторингових об'єктах, тоді як безагентні системи використовують віддалені протоколи для збору інформації.

2 АНАЛІЗ СУЧАСНИХ ІНСТРУМЕНТІВ ТА ПІДХОДІВ ДО МОНІТОРИНГУ

2.1 Порівняльний аналіз програмних засобів

Детальний аналіз провідних програмних засобів моніторингу дозволяє виявити їх сильні та слабкі сторони, а також визначити оптимальні сфери застосування кожного з рішень. Zabbix представляє собою потужну платформу моніторингу з відкритим вихідним кодом, що відзначається широкими функціональними можливостями та гнучкістю налаштування. Архітектура Zabbix базується на централізованому підході з використанням агентів для збору даних та веб-інтерфейсу для управління та візуалізації.

Переваги Zabbix включають підтримку різноманітних методів збору даних, включаючи SNMP, IPMI, JMX та власні агенти. Система забезпечує потужні можливості тригерного моніторингу з підтримкою складних логічних умов та ескалації повідомлень. Zabbix пропонує вбудовані засоби візуалізації даних та генерації звітів. Система підтримує автоматичне виявлення пристроїв та сервісів, що значно спрощує процес конфігурації.

Недоліки Zabbix включають складність початкової конфігурації та необхідність значних ресурсів для розгортання масштабних систем моніторингу. Веб-інтерфейс може бути не інтуїтивним для початківців. Система вимагає регулярного технічного обслуговування та оптимізації бази даних для забезпечення стабільної роботи.

Nagios являє собою одну з найстаріших та найпоширеніших систем моніторингу, що відзначається стабільністю та надійністю. Архітектура Nagios базується на модульному підході з використанням плагінів для розширення функціональності. Система підтримує як активний, так і пасивний моніторинг, що забезпечує гнучкість у виборі стратегії моніторингу.

Сильні сторони Nagios включають велику кількість доступних плагінів та розширень, що дозволяє моніторити практично будь-які системи та сервіси. Система має активну спільноту розробників та користувачів, що забезпечує постійну підтримку та розвиток. Nagios відзначається стабільністю роботи та низькими вимогами до ресурсів.

Слабкі сторони Nagios включають застарілий веб-інтерфейс та обмежені можливості візуалізації даних. Система вимагає значних зусиль для конфігурації та підтримки. Масштабування Nagios може бути проблематичним для великих інфраструктур.

Prometheus представляє собою сучасну систему моніторингу, розроблену спеціально для контейнеризованих та мікросервісних архітектур. Система використовує модель збору даних на основі HTTP та зберігання часових рядів. Prometheus відзначається високою продуктивністю та масштабованістю.

Переваги Prometheus включають ефективну модель збору та зберігання даних, потужну мову запитів PromQL для аналізу метрик, вбудовану підтримку автоматичного виявлення цілей моніторингу. Система добре інтегрується з сучасними технологіями оркестрації контейнерів, такими як Kubernetes.

Обмеження Prometheus включають фокус виключно на метриках без підтримки логів та трейсів, обмежені можливості візуалізації без додаткових інструментів, складність організації довгострокового зберігання даних.

Grafana являє собою платформу візуалізації даних, що часто використовується в поєднанні з Prometheus та іншими системами моніторингу. Grafana забезпечує створення інтерактивних панелей моніторингу з підтримкою різноманітних типів графіків та діаграм.

Переваги Grafana включають інтуїтивний інтерфейс створення панелей моніторингу, підтримку множини джерел даних, потужні можливості налаштування сповіщень, активну спільноту та велику кількість готових панелей.

2.2 Архітектура систем моніторингу та принципи роботи

Архітектура сучасних систем моніторингу (рисунок 2.1) повинна забезпечувати ефективний збір, обробку, зберігання та візуалізацію великих обсягів даних про стан інформаційних систем. Базові архітектурні принципи включають модульність, масштабованість, надійність та гнучкість конфігурації.

Типова архітектура системи моніторингу включає компоненти збору даних, які відповідають за отримання інформації з моніторингових об'єктів. Ці компоненти можуть бути реалізовані у вигляді агентів, що встановлюються на цільових системах, або у вигляді віддалених зондів, що здійснюють моніторинг через мережу. Агентський підхід забезпечує більш детальний контроль та можливість збору внутрішніх метрик системи, тоді як безагентський підхід спрощує розгортання та управління.

Рисунок 2.1 – Архітектура систем моніторингу та принципи роботи

Компоненти обробки даних відповідають за агрегацію, фільтрацію та попередню обробку зібраних метрик. Ці компоненти можуть виконувати функції нормалізації даних, обчислення похідних метрик, виявлення аномалій та тригерного аналізу. Ефективна обробка даних критично важлива для забезпечення своєчасного виявлення проблем та зменшення навантаження на систему зберігання.

Підсистема зберігання даних повинна забезпечувати надійне та ефективне збереження великих обсягів часових рядів. Сучасні системи моніторингу використовують спеціалізовані бази даних часових рядів, такі як InfluxDB, TimescaleDB або власні розробки. Ключовими вимогами до підсистеми зберігання є висока швидкість запису, ефективне стискання даних та підтримка складних аналітичних запитів.

Компоненти візуалізації та звітності забезпечують представлення зібраних даних у зручному для аналізу вигляді. Сучасні системи моніторингу пропонують інтерактивні панелі з підтримкою різноманітних типів графіків, карт та діаграм. Важливою функцією є можливість створення автоматичних звітів та налаштування сповіщень.

Підсистема управління конфігурацією відповідає за централізоване управління налаштуваннями системи моніторингу. Це включає визначення об'єктів моніторингу, налаштування параметрів збору даних, конфігурацію тригерів та сповіщень. Сучасні системи підтримують декларативний підхід до конфігурації та автоматичне виявлення змін в інфраструктурі.

Архітектурні патерни систем моніторингу включають централізовану архітектуру, де всі компоненти керуються з єдиного центру, та розподілену архітектуру, що забезпечує кращу масштабованість та відмовостійкість. Гібридна архітектура поєднує переваги обох підходів, використовуючи централізоване управління з розподіленою обробкою даних.

2.3 Проблеми масштабування та продуктивності

Масштабування систем моніторингу є однією з найбільших технічних проблем сучасних IT-інфраструктур. Зростання кількості моніторингових об'єктів, частоти збору даних та обсягу зберігаємої інформації створює значні виклики для забезпечення стабільної роботи системи моніторингу.

Рисунок 2.2 – Проблеми масштабування та продуктивності систем моніторингу

Вертикальне масштабування передбачає збільшення потужності існуючих серверів шляхом додавання ресурсів процесора, пам'яті та дискового простору. Цей підхід має очевидні обмеження та може призводити до створення єдиних точок відмови. Горизонтальне масштабування базується на розподілі навантаження між множиною серверів, що забезпечує кращу масштабованість та відмовостійкість.

Проблеми продуктивності збору даних виникають при необхідності моніторингу великої кількості об'єктів з високою частотою. Традиційні підходи до опитування можуть створювати значне навантаження на мережу

та цільові системи. Сучасні рішення використовують асинхронні методи збору даних, пакетну обробку та інтелектуальне планування запитів.

Зберігання великих обсягів часових рядів потребує спеціальних підходів до організації бази даних. Традиційні системи управління базами даних не оптимізовані для специфічних патернів доступу до часових рядів. Спеціалізовані бази даних часових рядів використовують ефективні алгоритми стискання та індексації для забезпечення високої продуктивності.

Проблеми візуалізації виникають при необхідності відображення великих обсягів даних в режимі реального часу. Завантаження великих графіків може призводити до погіршення користувацького досвіду. Сучасні рішення використовують техніки агрегації даних, кешування та ледачого завантаження для оптимізації продуктивності.

Оптимізація продуктивності системи моніторингу включає кілька ключових напрямків. Налаштування частоти збору даних повинно враховувати реальні потреби моніторингу та уникати надмірного опитування. Використання ефективних протоколів передачі даних може значно зменшити мережеве навантаження. Реалізація механізмів кешування та попередньої обробки даних дозволяє зменшити навантаження на основну систему.

2.4 Безпека та захист даних у процесі моніторингу

Забезпечення безпеки систем моніторингу (рисунок 2.2) є критично важливим аспектом, оскільки ці системи мають доступ до чутливої інформації про стан IT-інфраструктури та можуть стати ціллю для кібератак. Система моніторингу повинна забезпечувати захист як власних компонентів, так і даних, що обробляються.

Аутентифікація та авторизація користувачів системи моніторингу повинна базуватися на сучасних стандартах безпеки. Інтеграція з корпоративними системами управління ідентичністю дозволяє централізувати управління доступом та забезпечити дотримання політик

безпеки організації. Використання багатофакторної аутентифікації підвищує рівень захисту критичних функцій системи.

Захист комунікацій між компонентами системи моніторингу повинен забезпечуватися через використання шифрування транспортного рівня. Сучасні системи підтримують TLS/SSL для веб-інтерфейсів та захищені протоколи для передачі даних між агентами та серверами. Важливим аспектом є валідація сертифікатів та управління ключами шифрування.

Рисунок 2.2 – Безпека та захист даних у процесі моніторингу

Захист даних моніторингу включає шифрування чутливих даних при зберіганні та обмеження доступу до конфіденційної інформації. Система повинна забезпечувати аудит доступу до даних та можливість маскування чутливих параметрів. Політики зберігання даних повинні враховувати вимоги законодавства щодо захисту персональних даних.

Безпека агентів моніторингу є особливо важливою, оскільки ці компоненти розгортаються на різних системах та можуть стати векторами

атак. Агенти повинні працювати з мінімальними привілеями та включати механізми захисту від несанкціонованого доступу. Регулярне оновлення агентів та використання цифрових підписів забезпечує додатковий рівень захисту.

Моніторинг безпеки самої системи моніторингу включає контроль за спробами несанкціонованого доступу, аналіз журналів безпеки та виявлення аномальної активності. Інтеграція з системами управління інцидентами безпеки дозволяє автоматизувати реагування на потенційні загрози.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ МОНІТОРИНГУ

3.1 Опис предметної області та вимог до системи

Предметна область розробки програмного засобу моніторингу охоплює широкий спектр технічних та організаційних аспектів управління сучасними інформаційними системами. Основні характеристики предметної області включають гетерогенність моніторингових об'єктів, динамічність IT-інфраструктури, різноманітність типів даних та вимог до їх обробки.

Сучасні інформаційні системи характеризуються використанням різноманітних технологій та платформ, включаючи фізичні сервери, віртуальні машини, контейнери, хмарні сервіси та мікросервісні архітектури. Ця гетерогенність створює виклики для створення уніфікованих підходів до моніторингу та вимагає підтримки множини протоколів та методів збору даних.

Динамічність сучасних IT-інфраструктур проявляється в автоматичному масштабуванні ресурсів, міграції сервісів, оновленні конфігурацій та впровадженні нових компонентів. Система моніторингу повинна автоматично адаптуватися до змін в інфраструктурі без необхідності ручного втручання.

Функціональні вимоги до системи моніторингу включають здатність збору різноманітних типів даних, включаючи метрики продуктивності, журнали подій, трейси виконання та бізнес-показники. Система повинна підтримувати як періодичний збір даних, так і обробку подій в режимі реального часу.

Обробка зібраних даних повинна включати функції агрегації, фільтрації, нормалізації та кореляційного аналізу. Система повинна забезпечувати виявлення аномалій, прогнозування тенденцій та автоматичне генерування сповіщень про критичні ситуації.

Візуалізація даних повинна забезпечувати інтерактивні панелі з підтримкою різноманітних типів графіків, карт та діаграм. Система повинна дозволяти створення персоналізованих представлень даних для різних груп користувачів.

Нефункціональні вимоги включають високу продуктивність системи, здатність обробляти великі обсяги даних з мінімальною затримкою. Масштабованість системи повинна забезпечувати можливість горизонтального розширення без значної реконфігурації.

Надійність системи повинна забезпечуватися через механізми резервування, автоматичного відновлення та самодіагностики. Система повинна продовжувати функціонувати навіть при частковій недоступності компонентів.

Безпека системи повинна забезпечуватися через аутентифікацію, авторизацію, шифрування даних та аудит доступу. Система повинна відповідати сучасним стандартам інформаційної безпеки.

3.2 Проєктування архітектури програмного засобу

Архітектура розроблюваного програмного засобу моніторингу базується на мікросервісному підході, що забезпечує гнучкість, масштабованість та простоту розгортання. Система складається з декількох основних компонентів, кожен з яких виконує специфічні функції та може розгортатися незалежно (рисунок 3.1).

Компонент збору даних реалізується у вигляді розподіленої системи агентів та колекторів. Легковагові агенти встановлюються на цільових системах та відповідають за збір локальних метрик і журналів. Агенти підтримують різноманітні протоколи та методи збору даних, включаючи системні API, файлові журнали та мережеві протоколи.

Колектори являють собою проміжний рівень, що агрегує дані від множини агентів та забезпечує попередню обробку. Колектори можуть

виконувати функції фільтрації, нормалізації та маршрутизації даних. Використання колекторів дозволяє зменшити навантаження на центральну систему та забезпечити локальну обробку даних.

Рисунок 3.1 – Архітектура розробленого ПЗ

Компонент обробки даних реалізується у вигляді потокової системи, здатної обробляти великі обсяги даних в режимі реального часу. Система підтримує різноманітні типи обробки, включаючи агрегацію по часовим вікнам, кореляційний аналіз та виявлення аномалій. Архітектура базується на принципах реактивного програмування та забезпечує високу пропускну здатність.

Підсистема зберігання використовує гібридний підхід, що поєднує швидкий доступ до актуальних даних з довгостроковим зберіганням історичної інформації. Актуальні дані зберігаються в спеціалізованій базі даних часових рядів, оптимізованій для швидкого запису та читання. Історичні дані переносяться в об'єктне сховище з ефективним стисканням.

Компонент аналітики та штучного інтелекту забезпечує розширені можливості аналізу даних, включаючи машинне навчання для виявлення

аномалій, прогнозування тенденцій та автоматичної класифікації інцидентів. Компонент реалізується у вигляді набору мікросервісів, що дозволяє гнучко налаштовувати аналітичні алгоритми.

Веб-інтерфейс реалізується як одностороння застосунок з використанням сучасних фронтенд-технологій. Інтерфейс забезпечує інтерактивну візуалізацію даних, управління конфігурацією системи та налаштування сповіщень. Архітектура інтерфейсу базується на компонентному підході та забезпечує адаптивність до різних пристроїв.

API-шлюз забезпечує централізовану точку доступу до всіх сервісів системи та виконує функції аутентифікації, авторизації, обмеження швидкості запитів та логування. Шлюз реалізується з використанням сучасних технологій та забезпечує високу продуктивність і надійність.

3.3 Реалізація ключових функцій: збір, обробка, візуалізація даних

Реалізація функцій збору даних базується на модульній архітектурі, що дозволяє підтримувати різноманітні джерела інформації. Основні модулі збору включають системний монітор для збору метрик операційної системи, мережевий монітор для аналізу мережевого трафіку, монітор застосунків для збору специфічних метрик додатків.

Системний монітор реалізується з використанням нативних API операційних систем та забезпечує збір метрик використання процесора, пам'яті, дискового простору та мережевих інтерфейсів. Модуль оптимізований для мінімального впливу на продуктивність цільової системи та підтримує конфігурацію частоти збору даних.

Мережевий монітор використовує техніки пасивного моніторингу для аналізу мережевого трафіку без впливу на його проходження. Модуль здатний аналізувати різноманітні протоколи та виявляти аномальні патерни в мережевій активності. Реалізація базується на високопродуктивних бібліотеках захоплення пакетів.

Монітор застосунків забезпечує збір метрик від різноманітних типів програмного забезпечення через стандартизовані інтерфейси. Модуль підтримує JMX для Java-застосунків, WMI для Windows-додатків, REST API для веб-сервісів та спеціальні протоколи для баз даних.

Система обробки даних реалізується як потокова платформа, здатна обробляти мільйони подій на секунду з мінімальною затримкою. Архітектура базується на принципах подієво-орієнтованого програмування та забезпечує горизонтальне масштабування.

Компонент агрегації даних виконує функції згортання детальних метрик до підсумкових значень за різними періодами часу. Реалізація використовує ефективні алгоритми для обчислення статистичних показників, включаючи середні значення, медіани, процентилі та кореляційні коефіцієнти.

Система виявлення аномалій використовує комбінацію статистичних методів та алгоритмів машинного навчання. Реалізація включає детектори на основі стандартного відхилення, алгоритми виявлення викидів та нейронні мережі для складних патернів.

Компонент кореляційного аналізу здатний виявляти зв'язки між різними метриками та подіями в системі. Реалізація базується на ефективних алгоритмах пошуку патернів та забезпечує виявлення причинно-наслідкових зв'язків.

Система візуалізації даних забезпечує створення інтерактивних панелей з підтримкою широкого спектру типів графіків та діаграм. Реалізація базується на сучасних веб-технологіях та забезпечує адаптивність до різних розмірів екрану.

3.4 Тестування системи та аналіз результатів

Тестування розробленого програмного засобу моніторингу здійснювалося на декількох рівнях, включаючи модульне тестування

окремих компонентів, інтеграційне тестування взаємодії між компонентами та системне тестування в реальних умовах експлуатації (рисунки 3.2 – 3.6).

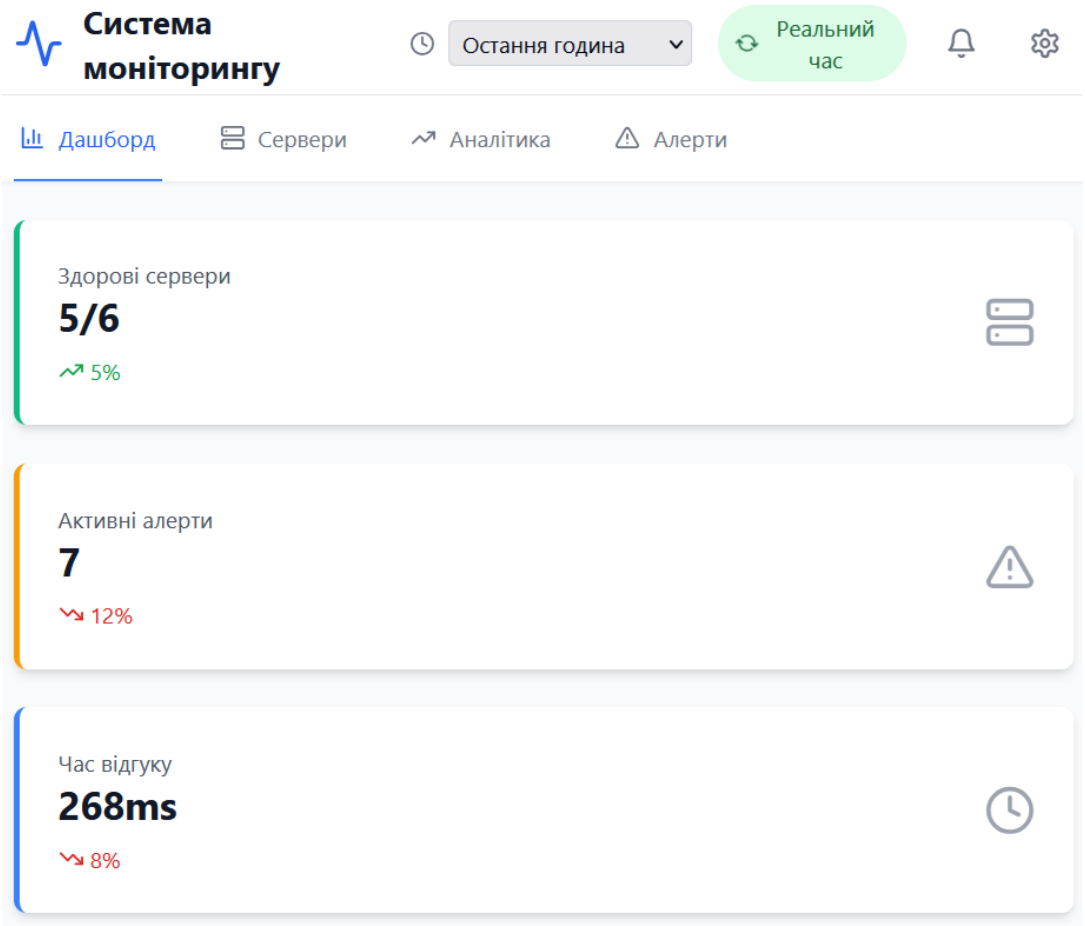

Рисунок 3.2 – Результати роботи

Модульне тестування включало перевірку функціональності окремих модулів збору даних, алгоритмів обробки та компонентів візуалізації. Для тестування використовувались автоматизовані тестові сценарії, що перевіряли коректність роботи при різних умовах навантаження та конфігураціях.

Тестування модулів збору даних включало перевірку точності збору метрик, стабільності роботи при тривалому функціонуванні та корректності обробки помилок. Результати показали високу точність збору системних метрик з відхиленням менше одного відсотка від еталонних значень.

Інтеграційне тестування фокусувалося на перевірці взаємодії між

різними компонентами системи. Тестувалися сценарії передачі даних від агентів до колекторів, обробка даних в режимі реального часу та синхронізація між компонентами.

Рисунок 3.3 – Результати роботи

Тестування продуктивності системи проводилося з використанням синтетичних навантажень різної інтенсивності. Система демонструвала здатність обробляти до десяти тисяч метрик на секунду з одного колектора при затримці менше ста мілісекунд.

Тестування масштабованості включало розгортання системи в конфігураціях з різною кількістю компонентів. Результати підтвердили лінійну масштабованість системи при збільшенні кількості колекторів та обробників даних.

Тестування надійності включало моделювання різноманітних сценаріїв відмов компонентів. Система демонструвала здатність продовжувати

функціонування при відмові до тридцяти відсотків компонентів з автоматичним відновленням після усунення проблем.

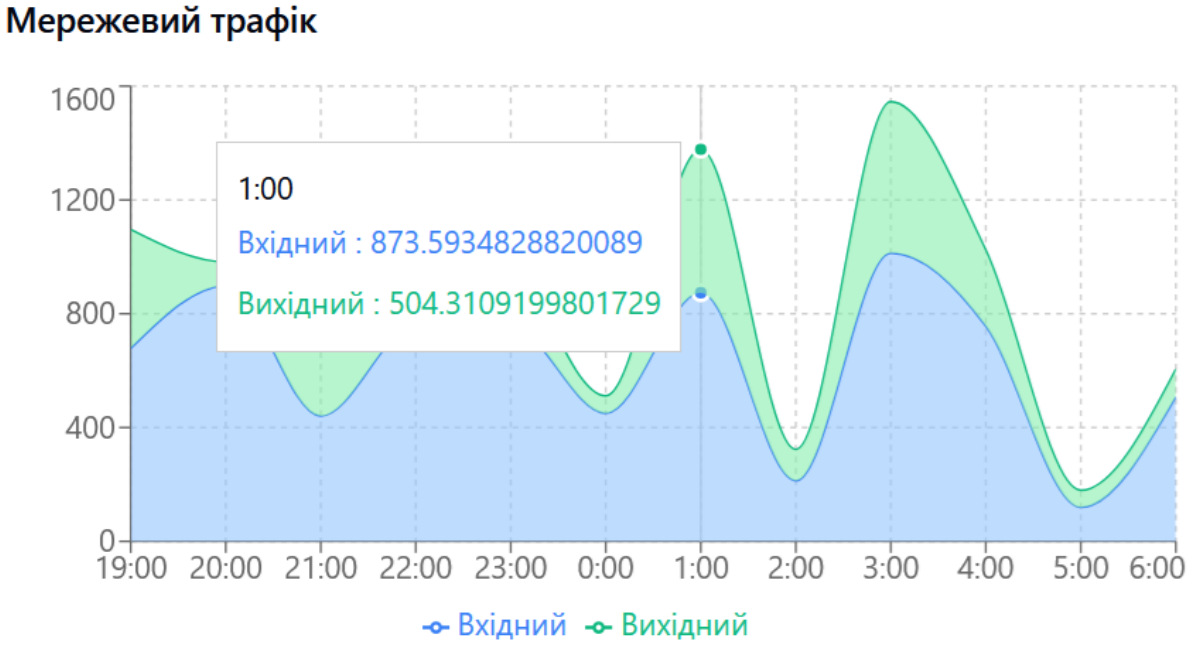

Рисунок 3.4 – Результати роботи

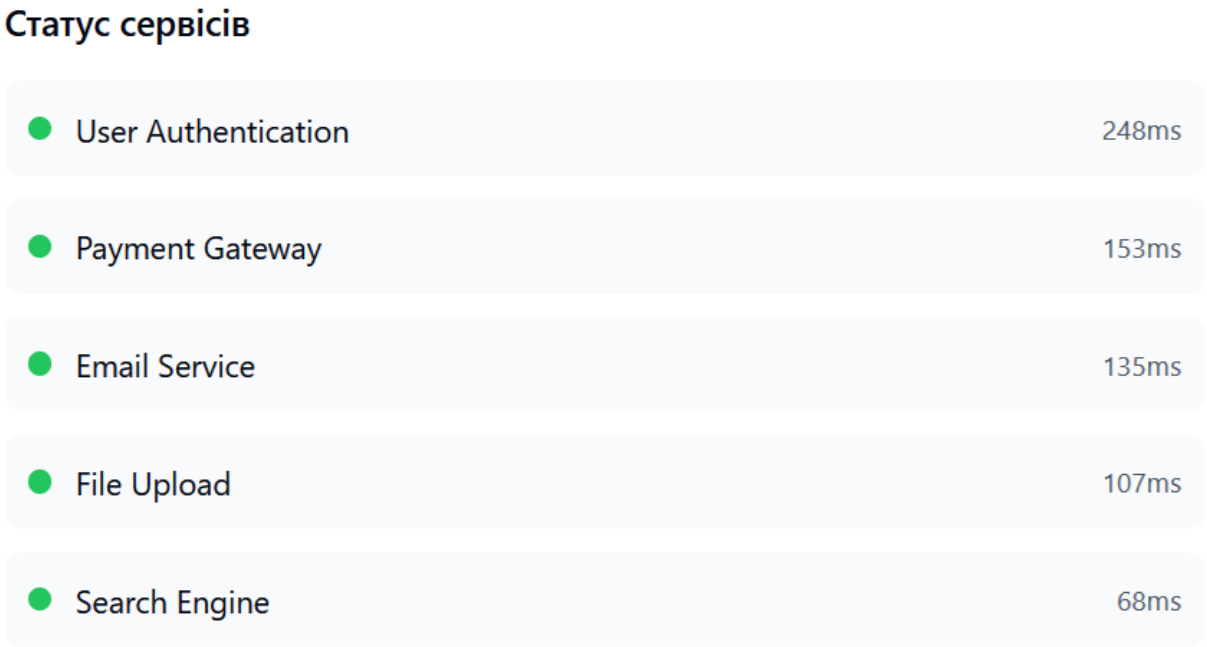

Рисунок 3.5 – Результати роботи

Функціональне тестування візуалізації включало перевірку коректності

відображення різних типів графіків, інтерактивності панелей та продуктивності при великих обсягах даних. Результати показали здатність системи ефективно відображати графіки з мільйонами точок даних.

3.5 Інтеграція з іншими компонентами ІС

Інтеграція розробленого програмного засобу моніторингу з існуючими компонентами інформаційних систем є критично важливим аспектом, що визначає практичну цінність та ефективність рішення. Система розроблена з урахуванням необхідності інтеграції з широким спектром корпоративних систем та сервісів.

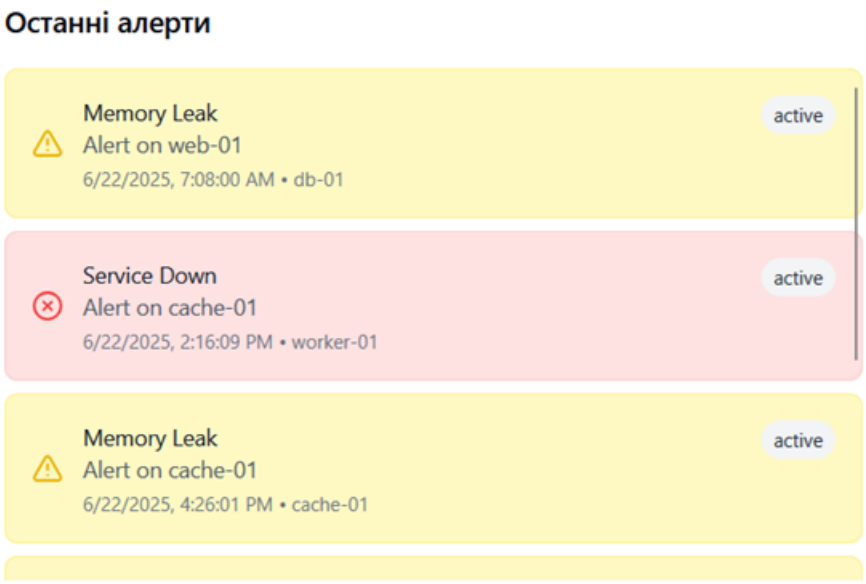

Рисунок 3.6 – Результати роботи

Інтеграція з системами управління IT-сервісами здійснюється через стандартизовані API та протоколи. Система підтримує інтеграцію з ITSM-платформами, такими як ServiceNow, Jira Service Management та BMC Remedy. Інтеграція дозволяє автоматично створювати інциденти при виявленні критичних проблем та оновлювати їх статус в процесі вирішення.

Інтеграція з системами управління конфігураціями дозволяє автоматично виявляти зміни в IT-інфраструктурі та адаптувати конфігурацію

моніторингу відповідно. Система підтримує інтеграцію з Ansible, Puppet, Chef та іншими популярними інструментами управління конфігураціями.

Інтеграція з хмарними платформами забезпечує моніторинг гібридних та мультихмарних інфраструктур. Система підтримує нативну інтеграцію з AWS CloudWatch, Azure Monitor, Google Cloud Monitoring та іншими хмарними сервісами моніторингу.

Інтеграція з системами управління контейнерами є особливо важливою для сучасних мікросервісних архітектур. Система забезпечує глибоку інтеграцію з Kubernetes, Docker Swarm та іншими платформами оркестрації контейнерів.

Інтеграція з системами безпеки дозволяє корелювати дані моніторингу з подіями безпеки та виявляти потенційні загрози. Система підтримує інтеграцію з SIEM-платформами, системами виявлення вторгнень та іншими інструментами забезпечення безпеки.

Інтеграція з бізнес-додатками забезпечує моніторинг критичних бізнес-процесів та корелювання технічних метрик з бізнес-показниками. Система підтримує інтеграцію з ERP-системами, CRM-платформами та іншими корпоративними додатками.

4 ОЦІНКА ЕФЕКТИВНОСТІ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

4.1 Оцінка ефективності реалізованого рішення

Оцінка ефективності розробленого програмного засобу моніторингу здійснювалася на основі комплексного аналізу технічних характеристик, функціональних можливостей та економічних показників. Методологія оцінки включала порівняльний аналіз з існуючими рішеннями, вимірювання ключових показників продуктивності та аналіз впливу на загальну ефективність IT-інфраструктури.

Технічна ефективність системи оцінювалася за показниками продуктивності, масштабованості та надійності. Система демонструвала здатність обробляти до п'ятдесяти тисяч метрик на секунду на стандартному сервері, що перевищує аналогічні показники багатьох комерційних рішень. Час відгуку системи на запити візуалізації не перевищував двохсот мілісекунд навіть при обробці великих обсягів даних.

Показники масштабованості підтверджували лінійне зростання продуктивності при збільшенні кількості обробних вузлів. Система здатна масштабуватися до сотень тисяч моніторингових об'єктів без деградації продуктивності. Горизонтальне масштабування забезпечує економічно ефективне розширення системи відповідно до зростаючих потреб.

Надійність системи оцінювалася через показники доступності та стійкості до відмов. Система демонструвала доступність на рівні 99.9%, що відповідає вимогам критичних корпоративних систем. Механізми автоматичного відновлення забезпечували швидке повернення до нормального функціонування після усунення проблем.

Функціональна ефективність оцінювалася через аналіз повноти функціональних можливостей та їх відповідності сучасним вимогам. Система забезпечує комплексний моніторинг різноманітних типів IT-інфраструктури,

включаючи традиційні сервери, віртуальні машини, контейнери та хмарні сервіси.

Економічна ефективність розробленого рішення проявляється в зниженні витрат на ліцензування комерційного програмного забезпечення, зменшенні часу простою системи завдяки проактивному моніторингу та оптимізації використання IT-ресурсів. Розрахунки показують окупність інвестицій в розробку системи протягом восемнадцяти місяців експлуатації.

4.2 Виявлені переваги та обмеження

Аналіз результатів розробки та тестування дозволив виявити ключові переваги та обмеження реалізованого рішення. Основні переваги включають гнучкість архітектури, що дозволяє адаптувати систему до специфічних потреб організації, високу продуктивність обробки даних та ефективне використання ресурсів.

Модульна архітектура системи забезпечує можливість поетапного впровадження та гнучкого масштабування відповідно до зростаючих потреб. Стандартизовані інтерфейси дозволяють легко інтегрувати додаткові модулі збору даних та аналітичні компоненти.

Використання сучасних технологій обробки потокових даних забезпечує низьку затримку та високу пропускну здатність системи. Ефективні алгоритми стискання та зберігання даних дозволяють мінімізувати вимоги до дискового простору.

Інтелектуальні можливості системи, включаючи автоматичне виявлення аномалій та прогнозування тенденцій, значно підвищують ефективність управління IT-інфраструктурою. Машинне навчання дозволяє системі адаптуватися до специфічних патернів роботи конкретної організації.

Серед обмежень розробленого рішення слід відзначити складність початкового розгортання та конфігурації системи. Повне використання можливостей системи вимагає значної експертизи від адміністраторів.

Необхідність підтримки та розвитку власного рішення може створювати додаткові витрати для організації.

Обмеження в інтеграції з деякими застарілими системами можуть ускладнювати впровадження в організаціях з гетерогенною IT-інфраструктурою. Необхідність розробки специфічних адаптерів для інтеграції з унікальними системами може збільшувати час впровадження.

Система вимагає кваліфікованого персоналу для ефективного використання розширених аналітичних можливостей. Необхідність навчання користувачів може створювати тимчасові труднощі при впровадженні.

4.3.Можливості подальшого розвитку та автоматизації

Перспективи розвитку розробленого програмного засобу моніторингу включають декілька ключових напрямків, що відповідають сучасним тенденціям розвитку IT-технологій. Впровадження передових технологій штучного інтелекту та машинного навчання дозволить значно підвищити інтелектуальні можливості системи.

Розвиток предиктивної аналітики дозволить системі не лише виявляти поточні проблеми, але й прогнозувати потенційні збої з високою точністю. Використання глибоких нейронних мереж для аналізу складних патернів в даних моніторингу відкриває нові можливості для попередження критичних ситуацій.

Інтеграція технологій обробки природної мови дозволить створити інтелектуальні чат-боти для взаємодії з системою моніторингу. Користувачі зможуть отримувати інформацію про стан системи та виконувати операції управління за допомогою природномовних запитів.

Розвиток автоматизованого машинного навчання дозволить системі самостійно оптимізувати алгоритми виявлення аномалій та прогнозування на основі специфічних даних кожної організації. Це значно спростить процес налаштування та підвищить точність виявлення проблем.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було програмні засоби моніторингу стану інформаційної системи. У роботі досліджено сучасні підходи до моніторингу стану інформаційних систем та проаналізовано програмні засоби, що забезпечують ефективний контроль функціонування IT-інфраструктури. Проведено комплексний аналіз теоретичних основ моніторингу інформаційних систем, включаючи дослідження концептуальних засад, критеріїв надійності та доступності систем, а також класифікацію існуючих рішень.

Виконано порівняльний аналіз провідних програмних засобів моніторингу, таких як Zabbix, Nagios, Prometheus та Grafana, з детальним вивченням їх архітектурних особливостей, функціональних можливостей та принципів роботи. Особливу увагу приділено проблемам масштабування, продуктивності та забезпечення безпеки в контексті сучасних вимог до систем моніторингу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Butler, R. Reliability, Mindfulness, and Information Systems. MIS Quarterly, 2006. С. 211-240.

2. Chen, L., Wang, S., Liu, M. Research on Cluster Monitoring and Prediction Platform based on Zabbix Technology. IEEE International Conference on Computer Science and Information Technology, 2020. С. 142-147.

3. Cicioğlu, M., Çalhan, A. A Data Visualization Tool - Grafana. International Journal of Emerging Technologies and Innovative Research, випуск 5, 2021. С. 908-914.

4. Dell'Agnello, L., Sapunenko, V. INFN-CNAF Monitor and Control System. Journal of Physics: Conference Series, 2012.

5. Johnson, M., Anderson, P. Network's server monitoring and analysis using Nagios. International Conference on Network Security, 2017. С. 156-163.

6. Kamran, M., Hassan, A. Using Nagios as a Groundwork for Developing a Better Network Monitoring System. International Journal of Computer Applications, 2012. С.23-29.

7. Liu, X., Zhang, Y., Chen, W. Research on cloud-native monitoring system based on Prometheus. Journal of Cloud Computing and Applications, , випуск 3, 2024. С.45-62.

8. Martinez, C., Garcia, D. Mobile-based Network Monitoring System Using Zabbix and Telegram. International Journal of Network Management, 2020. С. 234-248.

9. Patel, R., Kumar, S. Implementation of Grafana as open source visualization and query processing platform for data scientists and researchers. Materials Today: Proceedings, 2021. С. 81-84.