

ISSN 2307-5732

DOI 10.31891/2307-5732

НАУКОВИЙ ЖУРНАЛ

2.2024

ВІСНИК

**Хмельницького
національного
університету**

Технічні науки

Technical sciences

SCIENTIFIC JOURNAL

HERALD OF KHMELNYTSKYI NATIONAL UNIVERSITY

2024, Issue 2, Volume 333

Хмельницький

Редакційний штат

Головний редактор

Микола СКИБА

д. т. н., професор, голова Вченої ради Хмельницького національного університету, Лауреат Державної премії України в галузі науки і техніки, заслужений працівник народної освіти України, член-кореспондент Національної академії педагогічних наук України, Україна

Профіль Scopus: <https://www.scopus.com/authid/detail.uri?authorId=57204934613>

Заступник головного редактора

Олег СИНЮК

д. т. н., професор, проректор з наукової роботи Хмельницького національного університету, професор кафедри машин і апаратів, електромеханічних та енергетичних систем, м. Хмельницький, Україна.

Профіль Scopus: <https://www.scopus.com/authid/detail.uri?authorId=57196153541>

Відповідальний секретар

Сергій ГОРЯЩЕНКО

Горященко С. Л.

к.т.н., доцент, доцент кафедри машин і апаратів, електромеханічних та енергетичних систем, заслужений винахідник України, керівник Лабораторії енергозберігаючих технологій Хмельницького національного університету, м. Хмельницький, Україна.

Профіль Scopus: <https://www.scopus.com/authid/detail.uri?authorId=55855469200>

Члени редколегії серії «Технічні науки»

МЕТОД ПОШУКУ ЗАЛЕЖНОСТЕЙ У ВЕЛИКИХ НАБОРАХ ДАНИХ ІЗ ПРОПУСКАМИ НА ПРИКЛАДІ ДАНИХ ПРО ПОШИРЕННЯ COVID

НАТАЛІЯ ШАХОВСЬКА, ІВАН ЗАГОРОДНИЙ (Автор)

11-16

[PDF](#)

СИСТЕМА АВТОМАТИЗОВАНОГО РОЗРАХУНКУ ПАРАМЕТРІВ ЕТАПУ ЗАПУСКУ БЕЗПІЛОТНОГО ЛІТАЛЬНОГО АППАРАТУ З ПУСКОВОЇ УСТАНОВКИ КАТАПУЛЬТНОГО ТИПУ

ДІАНА БУКОВСЬКА, ВІКТОР АНТОНЮК (Автор)

17-20

[PDF \(Англійська\)](#)

ФОРМУВАННЯ СПОЖИВНИХ ВЛАСТИВОСТЕЙ ЕЛАСТИЧНИХ ШКІРЯНИХ МАТЕРІАЛІВ

АНАТОЛІЙ ДАНИЛКОВИЧ, ВІКТОР ЛІЩУК (Автор)

21-27

[PDF](#)

ПОБУДОВА КОДІВ ХЕММІНГА В СКІНЧЕННИХ ПОЛЯХ ГАЛУА

АЛІНА ДАВЛЕТОВА (Автор)

28-34

[PDF](#)

ПІДХОДИ ДО АВТОМАТИЗАЦІЇ ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Роман Феняк, Виклюк Ярослав Ігорович (Автор)

35-39

[PDF](#)

ПАРАЛЕЛЬНИЙ АНАЛІЗ МЕДИЧНИХ ТЕРМОГРАФІЧНИХ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ МЕТОДУ ОПОРНИХ ВЕКТОРІВ ТА НАЇВНОГО БАЙЄСІВСЬКОГО КЛАСИФІКАТОРА

Леся Гентош, Юліанна Мочурад, Дарина Василяшко (Автор)

40-45

[PDF](#)

ПОШИРЕННЯ ТА ВІДБИТТЯ ЗВУКОВОЇ ХВИЛІ НА ОСНОВІ ШУМУ ТРАМВАЮ

МАРІЯ ОРИНЧАК (Автор)

46-49

[PDF](#)

ІНТЕРФЕЙСИ ВІРТУАЛЬНИХ СХОВИЩ ДАНИХ В УМОВАХ ШВИДКОЗМІНЮВАНОВОГО ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА

НАТАЛІЯ ШАХОВСЬКА, ІВАН ЗАГОРОДНИЙ, ОКСАНА КОГУЧ, РОМАН ТРИСКА (Автор)

50-54

[PDF](#)

ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ АКАДЕМІЧНОЇ БІБЛІОТЕКИ

МАРІЯ СОКІЛ, АНДРІЙ ЗВОРСЬКИЙ (Автор)

55-60

[PDF](#)

РОЗВ'ЯЗАННЯ ЗАДАЧІ ДИНАМІЧНОЇ РЕКОНФІГУРАЦІЇ РОЗПОДІЛЬНОЇ МЕРЕЖІ З ДЖЕРЕЛАМИ РОЗПОДІЛЬНОЇ ГЕНЕРАЦІЇ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

ДМИТРО ЯЦЕНКО, ВОЛОДИМИР ПОПОВ, ВАДИМ КОНДРАТЮК, АНАТОЛІЙ ЗАМУЛКО, ІВАН ФРОЛОВ (Автор)

61-69

[PDF](#)

ДЕФОРМУВАННЯ ТА РУЙНУВАННЯ МАТЕРІАЛІВ З СУТТЄВИМ ПРОЯВОМ ВНУТРІШНЬОГО ТЕРТЯ

ОЛЕНА БАГРІЙ, ОЛЕКСАНДР ДОРОФЄЄВ (Автор)

78-84

[PDF](#)

СХЕМА ТА ТЕОРЕТИЧНЕ ОБГРУНТУВАННЯ ДІЄВОСТІ РЕЗОНАНСНОГО ПІДСИЛЮВАЧА АКТИВНОЇ ЕЛЕКТРИЧНОЇ ПОТУЖНОСТІ

ЮРІЙ БАТИГІН, ОЛЕНА ЄРЬОМІНА, СВІТЛАНА ШИНДЕРУК, ЄВГЕН ЧАПЛИГІН (Автор)

85-91

[PDF](#)

ВИКОРИСТАННЯ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ ДЛЯ СЕГМЕНТАЦІЇ ПЕРСОНАЛУ КОМПАНІЇ

БОГДАН БОЙКО, ІРИНА ПРОЦИК (Автор)

92-98

[PDF](#)

ВИКОРИСТАННЯ МЕТОДІВ ТА МОДЕЛЕЙ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ПОКРАЩЕННЯ ЕКСПЕРТНИХ СИСТЕМ ВІЯВЛЕННЯ ВТОРГНЕНЬ

**ШВИДКОДІЙНИЙ ПРИСТРІЙ АНАЛОГО-ЦИФРОВОГО ПЕРЕТВОРЕННЯ СИГНАЛІВ З ЦИФРОВИМ КОРИГУВАННЯМ ПОХИБОК ЧАСОВОЇ НЕВИЗНАЧЕНОСТІ**

ГЕННАДІЙ БОРТНИК, СЕРГІЙ БОРТНИК, МИХАЙЛО БРИЛЬ, ОЛЕКСАНДР БОРТНИК (Автор)

107-111

**АНТИБАКТЕРІАЛЬНІ АГЕНТИ ДЛЯ СТВОРЕННЯ ТЕКСТИЛЬНИХ МАТЕРІАЛІВ – СУЧАСНИЙ СТАН ТА ТЕНДЕНЦІЇ РОЗВИТКУ**

АННА ВАРДАНЯН, ЯНА РЕДЬКО (Автор)

112-119

**ДОСЛІДЖЕННЯ ЧИСЛОВОЇ ВИПАДКОВОЇ ПОСЛІДОВНОСТІ, ЩО ОДЕРЖАНА З ВЕБ КАМЕРИ**

ДМИТРО ГАНЖЕЛО, ГЕОРГІЙ ПРОХОРОВ (Автор)

120-124

**ДОСЛІДЖЕННЯ СТРАТЕГІЇ БАГАТОЕТАПНОГО РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКЛЮЧЕННЯМ СТОРОННІХ ЕФЕКТИВ**

ВІТАЛІЙ СУПРИГАН (Автор)

125-150

**МЕТОДИ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ З ВИРАЖЕНОЮ СЕЗОННІСТЮ НА ОСНОВІ ТРАНСФОРМЕРІВ**

КИРИЛО ЄМЕЦЬ (Автор)

131-134

**АКТУАЛЬНІСТЬ ВИРОБНИЦТВА СПЕЦІАЛЬНОГО ВЗУТТЯ ДЛЯ АГРЕСИВНОГО СЕРЕДОВИЩА**

ЛЮДМИЛА КОЗЛОВСЬКА, ОЛЕГ ПОМАРАНСЬКИЙ (Автор)

135-139

**АВТОМАТИЗОВАНІ СИСТЕМИ КЕРУВАННЯ ДОСТУПОМ**

ВОЛОДИМИР КОРЧИНСЬКИЙ, ІРИНА ТАРАСЕНКО, СЕРГІЙ РАЦИБОРИНСЬКИЙ, ОЛЕКСАНДР АКАЄВ, АРТЕМ ХАДЖИОГЛО (Автор)

140-145

**ОГЛЯД МОЖЛИВОСТІ ПОКРАЩЕННЯ АВТОМАТИЗОВАНОГО ПІДБОРУ NS КОДУ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ОПТИМІЗАЦІЇ МИТНОГО КЛАСИФІКАЦІЙНОГО ПРОЦЕСУ**

ЮРІЙ КРИВЕНЧУК, СТЕПАН КРУПА (Автор)

146-149

**ПРОЦЕС ЗБОРУ ТА ПІДГОТОВКИ ДАНИХ З ДЕМОНСТРАЦІЙ, ЗРОБЛЕНИХ ЛЮДИНОЮ, ДЛЯ ВИКОНАННЯ ПРОСТИХ МАНІПУЛЯЦІЙНИХ ЗАВДАНЬ ЗА ДОПОМОГОЮ РОБОРУКИ XARM7 ДЛЯ НАВЧАННЯ З ДОПОМОГОЮ BEHAVIOR CLONING АЛГОРИТМУ**

ЮРІЙ КРИВЕНЧУК, МАКСИМ ШАВАРСЬКИЙ (Автор)

150-154

**ВСТАНОВЛЕННЯ ВПЛИВУ ТИПУ ТЕКСТУРОУТВОРЮЮЧИХ ЧАСТИНОК НА ГІДРОФОБНІ ВЛАСТИВОСТІ ПОКРИТТЯ**

ЛІ ЧЕ, АННА БІЛОУСОВА, ОЛЕКСІЙ МИРОНЮК (Автор)

155-161

**ІНФОРМАТИЗАЦІЯ ТА УПРАВЛІННЯ НАУКОВИМИ ДОСЛІДЖЕННЯМИ: АЛГОРИТМ ПОБУДОВИ РЕЙТИНГУ ПУБЛІКАЦІЙНОЇ АКТИВНОСТІ**

ГАННА ЛІХОНОСОВА (Автор)

162-165

**ЗАСТОСУВАННЯ Soft Open Point ДЛЯ ОПТИМІЗАЦІЇ ВТРАТ ЕЛЕКТРОЕНЕРГІЇ В РОЗПОДІЛЬНИХ ЕЛЕКТРИЧНИХ МЕРЕЖАХ**

ЮРІЙ ЛОБОДА, РУСЛАН СЛОБОДЯН (Автор)

166-170

**АСПЕКТИ РЕАГУВАННЯ НА ІНЦИДЕНТИ ЗУМОВЛЕНІ АНОМАЛІЯМИ ФУНКЦІОНУВАННЯ ТА ОБСЛУГОВУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ**

**МОДЕЛІ ВІЯВЛЕННЯ АНОМАЛІЙ ДЛЯ СЕНСОРНИХ ДАНИХ ПРОЦЕСУ БУРІННЯ НАФТОГАЗОВИХ СВЕРДЛОВИН В УМОВАХ НЕВИЗНАЧЕНОСТІ**

ВОЛОДИМИР ПРОЦЬЮК (Автор)

177-188

**ВІЯВЛЕННЯ ШАХРАЙСТВА З КРЕДИТНИМИ КАРТКАМИ МЕТОДАМИ МАШИННОГО НАВЧАННЯ**

КАТЕРИНА МЕЛЬНИК, СВІТЛАНА ЛАВРЕНЧУК, НАТАЛІЯ ХРИСТИНЕЦЬ (Автор)

189-193

**ОСОБЛИВОСТІ ФОРМУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗВИТКУ ІНДИВІДУАЛЬНОЇ ТРАЕКТОРІЇ В ЗАКЛАДАХ ОСВІТИ**

ІГОР МІРКО (Автор)

194-199

**МЕТОД НЕЙРОМЕРЕЖЕВОГО ВІЯВЛЕННЯ КІБЕРБУЛІНГУ З ВИКОРИСТАННЯМ ХМАРНИХ СЕРВІСІВ ТА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ МОДЕЛІ**

МАРИНА МОЛЧАНОВА, ОЛЕКСАНДР МАЗУРЕЦЬ, ОЛЕНА СОБКО, ВАЛЕРІЯ КЛІМЕНКО, ВЛАДИСЛАВ АНДРОЩУК (Автор)

200-206

**ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ GAN МЕРЕЖ ДЛЯ ПОКРАЩЕННЯ НАДІЙНОСТІ ДАНИХ З ДАВАЧІВ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ**

Дмитро Петренко, Юрій Кривенчук (Автор)

207-211

**ПОРІВНЯННЯ РІЗНИХ ВЕРСІЙ ДОДАТКУ Midjourney ТА ЙОГО ВИКОРИСТАННЯ В РІЗНИХ СФЕРАХ СТВОРЕННЯ ВІЗУАЛЬНОГО КОНТЕНТУ**

ДМИТРО ПЕТРИНА, ОЛЕНА КОРНУТА (Автор)

212-217

**АНСАМБЛЕВІ МОДЕЛІ ПРОГНОЗУВАННЯ ПОВЕНЕЙ У ВЕЛИКІЙ БРИТАНІЇ НА ОСНОВІ СОНЯЧНОЇ АКТИВНОСТІ**

ПЕТРО СИДОР, ЯРОСЛАВ ВИКЛЮК (Автор)

218-231

**ТОВАРОЗНАВЧА ЕКСПЕРТИЗА МАРКУВАННЯ КОСМЕТИЧНИХ ТОВАРІВ**

СВІТЛАНА СІРЕНКО, АЛЛА ТЕРНОВА, ІРИНА ВЛАСЕНКО (Автор)

232-237

**ТЕХНОЛОГІЇ ОДЕРЖАННЯ ПОЛІМЕРНИХ КОМПОЗИЦІЙ ДЛЯ ВИГОТОВЛЕННЯ ПОГОНАЖНИХ ВИРОБІВ**

ОКСАНА СМАЧИЛО (Автор)

238-241

**ВІД АНАЛІЗУ ТЕКСТУ ДО МОДЕЛЮВАННЯ ПРИРОДНОЇ МОВИ: КОМПЛЕКСНЕ ДОСЛІДЖЕННЯ**

МИКОЛА СТАХІВ, СТАПАН СКОПІВСЬКИЙ (Автор)

242-250

**ПРОБЛЕМИ ЗАСВОЄННЯ БАЗОВИХ ДИСЦИПЛІН СТУДЕНТАМИ ВНЗ ТЕХНІЧНИХ НАПРЯМІВ ПІДГОТОВКИ В УМОВАХ ВІЙСЬКОВОГО СТАНУ УКРАЇНИ**

ЛЕСЯ СТРУТИНСЬКА (Автор)

251-259

**СТВОРЕННЯ ТЕПЛОЕФЕКТИВНИХ КЕРАМІЧНИХ ВИРОБІВ З МІСЦЕВОЇ ЛЕГКОПЛАВКОЇ СИРОВИНИ**

ІРИНА СУББОТА, ЛАРИСА СПАСЬОНОВА (Автор)

260-264

**ПОРІВНЯННЯ МЕТОДІВ ВІЯВЛЕННЯ ОБ'ЄКТІВ В КОМП'ЮТЕРНОМУ ЗОРІ**

ПАВЛО ЦИВАДИЦЬ, ТЕТЯНА СКРИПНИК, ЛЕОНІД ВОЗНЮК (Автор)

265-268

**ЦИФРОВІЗАЦІЯ ВИРОБНИЧИХ ТА БІЗНЕС-ПРОЦЕСІВ В АГРАРНОМУ СЕКТОРІ ЕКОНОМІКИ**



КОМП'ЮТЕРНИЙ МОДУЛЬ ПРОГРАМИ ДЛЯ ВИЗНАЧЕННЯ ОПТИМАЛЬНОЇ ФОРМИ ЗАПРАВКИ НИТКИ НА ТЕХНОЛОГІЧНОМУ ОБЛАДНАННІ

ВОЛОДИМИР ЩЕРБАНЬ, МИКИТА КОЛЬВА, ДМІТРО ЄГОРОВ, СЕРГІЙ ШИЛІНГОВ, АНДРІЙ ЛУКАШЕВ (Автор) 274-277



ОСОБЛИВОСТІ СТИСНЕННЯ З ВТРАТАМИ ЗОБРАЖЕНЬ ІЗ ШУМОМ

СЕРГІЙ КРИВЕНКО, ВОЛОДИМИР ЛУКІН (Автор)

278-283



ОСОБЛИВОСТІ СТРУКТУРИ БАЗИ ДАНИХ ДЛЯ ЗБЕРЕЖЕННЯ РІЗНОТИПНОЇ ІНФОРМАЦІЇ В СИСТЕМАХ КОМПЛЕКСНОГО МОНІТОРИНГУ

ЛЮБОВ ПОЛЯГУШКО, ВОЛОДИМИР СЛІПЧЕНКО, ОЛЬГА КРУШ, ВОЛОДИМИР РУДИК (Автор)

284-290



РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ЦИФРОВОЇ КАРТКИ ДОМАШНЬОЇ ТВАРИНИ

ІЛЛЯ СОЛОВЕЙ, ОЛЬГА ВОРОЧЕК (Автор)

291-296



ДВОКАНАЛЬНИЙ ГРАВІМЕТР АВІАЦІЙНОЇ ГРАВІМЕТРИЧНОЇ СИСТЕМИ

ОЛЕНА БЕЗВЕСІЛЬНА, МАРІЯ ГРИНЕВИЧ, ТЕТЯНА ТОЛОЧКО (Автор)

297-301



АСПЕКТИ КЛАСИФІКАЦІЇ ПРИСТОЇВ ТА СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ ПРИ ЇХ ІМПЛЕМЕНТАЦІЇ

ВАСИЛЬ БОДНАРУК, ІРИНА МАНУЛЯК, СТЕПАН МЕЛЬНИЧУК (Автор)

302-307



РОЗРОБКА ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ВИБОРУ СТИЛЮ ОДЯГУ НА ОСНОВІ KANSEI ENGINEERING

СВІТЛАНА КУЛЕШОВА (Автор)

308-313



ПІДВИЩЕННЯ АНТИФРИКЦІЙНИХ ХАРАКТЕРИСТИК ПОРШНЕВИХ УЩІЛЬНЕНЬ КОМПРЕСОРА КОНДИЦІОНЕРА АВТОМОБІЛЯ

ОЛЕКСАНДР ДИХА, ВЛАДИСЛАВ СВИДЕРСЬКИЙ, КОСТЯНТИН ГОЛЕНКО (Автор)

314-321



ПАТЕРНОВИЙ АНАЛІЗ ГРАФА БІТКОЇН-ТРАНЗАКЦІЙ

Жикін Юрій Сергійович, Онай Микола Володимирович (Автор)

322-328



ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ ДЛЯ МОНІТОРИНГУ ФІЗИЧНИХ НАВАНТАЖЕНЬ ВЕЛОСПОРТСМЕНІВ

Ірина Журавська, Богдан Пилипчук (Автор)

329-337



РАДІОВИМІРЮВАЛЬНИЙ ПЕРЕТВОРЮВАЧ ВОЛОГОСТІ ҐРУНТУ НА БАЗІ ПЕРВИННОГО СЕНСОРА YL-69

ЄВГЕНІЙ КРОПИВ'ЯНСЬКИЙ, ОЛЕКСАНДР ЗВЯГІН (Автор)

338-343



АНАЛІЗ ВИМОГ ДО ПОЗИЦІЙ DATA ANALYST ТА DATA SCIENTIST НА РИНКУ ПРАЦІ

ВОЛОДИМИР КУЛАЖЕНКО, ДЕНИС ПІДГАЙНИЙ (Автор)

344-352



ДИЗАЙН-ПРОЄКТУВАННЯ КОРПОРАТИВНОГО ОДЯГУ: ЕСТЕТИЧНИЙ ТА ЕРГОНОМІЧНИЙ АСПЕКТИ

СВІТЛАНА КУЛЕШОВА, ЮЛІЯ КОШЕВКО, ІРИНА КРИВИЦЬКА, ОКСАНА ЗАХАРКЕВИЧ, В'ЯЧЕСЛАВ БАЛАБАНОВ (Автор)

353-360



ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПОСТКВАНТОВОГО ЦИФРОВОГО ПІДПИСУ

ІВАН ЛАВРИК (Автор)

361-369



ПОРІВНЯЛЬНИЙ АНАЛІЗ РОБІТ ІЗ ДОСЛІДЖЕННЯ ПОВЕДІНКИ ТВАРИН ЗА ЇХНІМИ ЗВУКАМИ

Андрій Михайлів (Автор)

370-375

[PDF](#)

ВИКОРИСТАННЯ СУЧАСНИХ ТЕХНОЛОГІЙ У ДОСЛІДЖЕННІ МЕХАНІЧНИХ ХАРАКТЕРИСТИК ТЕКСТИЛЬНИХ НИТОК

ОЛЕГ НАХАЙЧУК, ЕЛІНА ЗАХАРОВА, ВАЛЕНТИНА ГОРОБЧИШИНА, ОКСАНА ЦИРУЛЬНИК (Автор)

376-379

[PDF](#)

СЕНСОРИ ТЕМПЕРАТУРИ НА БАЗІ CMOS

ВОЛОДИМИР МАРТИНЮК, ОЛЕКСАНДР МАЛЮК (Автор)

380-388

[PDF](#)

СУЧАСНІ МЕТОДИ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ВОЛОДИМИР ПЕДЯШ, ЄВГЕН ЛЕДОВСЬКИЙ, ВОЛОДИМИР ТКАЧ (Автор)

389-392

[PDF](#)

РОЗРОБКА КОНВЕКТИВНО-ВІБРАЦІЙНОЇ СУШАРКИ ДЛЯ СУШІННЯ ВОЛОСЬКИХ ГОРІХІВ

ОЛЕГ ЦУРКАН, АНАТОЛІЙ СПІРІН, ВОЛОДИМИР РУТКЕВИЧ, АНДРІЙ ДІДИК (Автор)

393-399

[PDF](#)

МЕТОДИКА ВИЗНАЧЕННЯ ХІМІЧНОЇ СТІЙКОСТІ ТА ПРОНИКНОСТІ ПРОБ СПЕЦІАЛЬНИХ МАТЕРІАЛІВ ДЛЯ ЗАХИСНОГО ОДЯГУ ДО ВПЛИВУ АМІАКУ

НІНА МИХАЙЛОВА, ВАЛЕРІЙ ПРИВАЛА (Автор)

400-403

[PDF](#)

ФІЗИЧНА МОДЕЛЬ ВЗАЄМОДІЇ СТРУМЕНІВ ПОВІТРЯ З РЕЛЬЄФОМ ПЛОСКОЇ ПОВЕРХНІ

СЕРГІЙ ПУНДИК, АНАТОЛІЙ КАРМАЛІТА (Автор)

404-408

[PDF](#)

ОГЛЯД ТА АНАЛІЗ ІНФОРМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ЯК ОСНОВ СУЧАСНОГО ПРОГРАМУВАННЯ

ОЛЬГА КРАВЧУК (Перекладач); АНДРІЙ КРАВЧУК (Автор)

409-413

[PDF](#)

ОПТИМІЗАЦІЯ КОНСТРУКЦІЇ ТА РЕЖИМУ РОБОТИ СУШИЛЬНОЇ КАМЕРИ НА ОСНОВІ ЇЇ 3D МОДЕЛІ

ТЕТЯНА OLYANYSHEN, ОЛЕГ МАЧУГА (Автор)

414-420

[PDF \(Англійська\)](#)

ДОСЛІДЖЕННЯ АЛГОРИТМУ ПОБУДОВИ МОДЕЛІ СЕНТИМЕНТ АНАЛІЗУ ПОВІДОМЛЕНЬ У СОЦІАЛЬНИХ МЕРЕЖАХ

ВОЛОДИМИР КУЛАЖЕНКО, ОЛЕКСАНДРА МАЗУР (Автор)

421-427

[PDF](#)

МАТЕМАТИЧНА МОДЕЛЬ ОЦІНКИ НАДІЙНОСТІ ДЛЯ ДІАГНОСТУВАННЯ ТЕХНІЧНОГО СТАНУ ЕЛЕКТРОТЕХНІЧНИХ КОМПЛЕКСІВ РОЗПОДІЛЕНИХ СИСТЕМ ВОДОПОСТАЧАННЯ

АНДРІЙ ТКАЧУК, МИКОЛА МОШНОРІЗ (Автор)

428-432

[PDF](#)

ВИКОРИСТАННЯ МОДУЛІВ КОМП'ЮТЕРНОЇ ПРОГРАМИ K DAM ДЛЯ ДИНАМІЧНОГО АНАЛІЗУ БАТАННОГО МЕХАНІЗМУ ПРИ ФОРМУВАННІ БАГАТОШАРОВИХ ТКАНИН

ВОЛОДИМИР ЩЕРБАНЬ, ОЛЕКСІЙ ВОЛЯНИК, МАР'ЯНА ГОЛЬДБЕРГ, ГЕННАДІЙ МЕЛЬНИК, ЮРІЙ ЩЕРБАНЬ (Автор)

433-436

[PDF](#)

ПОКРАЩЕННЯ ДЕТЕРМІНОВАНОГО МЕТОДУ ГЕНЕРУВАННЯ КОРПУСІВ ТЕКСТОВИХ ДАНИХ

ЯКІВ ЮСИН, НАТАЛІЯ РИБАЧОК (Автор)

437-445

[PDF](#)

СУПРИГАН ВІТАЛІЙ

Вінницький національний аграрний університет

<https://orcid.org/0000-0003-2539-8003>e-mail: suprigan@gmail.com

ДОСЛІДЖЕННЯ СТРАТЕГІЇ БАГАТОЕТАПНОГО РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКЛЮЧЕННЯМ СТОРОННІХ ЕФЕКТІВ

Стаття описує проблему та її рішення, яка виникла під час додавання нових користувачів у існуючий сервіс імпортування та розсилання даних. Проведено аналіз усіх доступних рішень та показано, що не існує такого рішення, яке може повністю задовольнити вимогам. Запропоновано багатокрокову стратегію доставки рішення, що дозволила сервісу відповідати вимогам доступності. Науковою новизною статті є опис та дослідження нового способу доставки рішення у середовище розгортання програмного забезпечення – production.

Ключові слова: high availability, continuous integration, continuous delivery, continuous deployment, git flow, MongoDB, індекси баз даних, бізнес-модель за підпискою.

SUPRYHAN VITALII

Vinnytsia National Agrarian University

STUDY OF MULTI-STAGE SOFTWARE DEPLOYMENT STRATEGY WITH THE EXCLUSION OF SIDE EFFECTS

The article addresses a problem and its solution that arose during onboarding of new subscribers to the existing data import and distribution service. The analysis of all available solutions demonstrates that none fully meet the requirements. The process of searching for the values of the required indicators and the process of creating reports necessarily uses indexes. Messages of different types are placed in their respective MongoDB collections. Over the long period of operation of the repository, up to 64 indexes per collection have accumulated. According to the policies of connecting to the storage (Connect timeout) and waiting for a response (Socket timeout), the need to set such long-time expectations for consumers is unacceptable and usually, by default, is equal to 30 seconds.

A multi-step strategy for delivering the solution that allows the service to meet availability requirements was proposed. The novelty of the article is in the description and investigation of a new method of delivering solutions to the software deployment environment—production. The implementation of the described strategy opens a new approach for solving similar classes of problems related to the type of interaction with users, when custom settings are a mandatory part of the service and are guaranteed to carry the threat of service unavailability. This approach opens up new opportunities for achieving high availability of systems, reducing risks and increasing user satisfaction. Based on the analysis of the proposed approaches, the latest strategy of multi-step delivery of the service to users was developed, the result of which was the acceptable response waiting time of up to 5 seconds. The implementation of this strategy contributed to the achievement of the availability of software products - 99.999%, ensuring the reliability and stability of services for end users.

Keywords: high availability, continuous integration, continuous delivery, continuous deployment, git flow, MongoDB, database indexes, subscription business model.

Вступ

У сучасному світі, де бізнес-процеси стають все більше залежними від інформаційних технологій, питання високої доступності систем набуває особливої актуальності. Висока доступність — це концепція, спрямована на забезпечення безперервної роботи системи або сервісу, мінімізацію часу простою та забезпечення надійного доступу до ресурсів для користувачів та бізнесу в будь-який час [1].

Мета дослідження – вирішити практичну задачу як уникнути блокування доставки інформації підписникам та узагальнити рішення для повторного використання в галузі.

Актуальність дослідження полягає у тому, що компанії потребують сучасних методик, які гарантовано створюють послідовності дій для вирішення проблем із допустимими та детермінованими експлуатаційними показниками на етапах: модифікації сервісу та обслуговування клієнтів.

Опис проекту. В структурі банку є підрозділ, який накопичує та оновлює фінансові показники. Такі показники надходять з різних джерел і заміщують собою попередні значення. Протягом робочого дня середня швидкість надходження неструктурованих повідомлень складає 600 Мбайт/хв. На поточний момент часу використовується сховище MongoDB, як таке що відповідає вимогам швидкодії та відсутності вимог до структури даних. Задача сервісу – сформувати звіти для користувачів та вивантажити їх по запланованій процедурі. Також користувачі можуть самі відправляти запити у сховище щоби наживо отримувати актуальні показники. Для обох випадків: для вивантаження та для запитів, розроблений API взаємодії як для поточних так і для майбутніх постачальників та споживачів показників.

Постановка проблеми

Суть проблеми полягає у тому, що клієнт намагається дізнатися які типи даних він потребує та створює заявку на обслуговування. Процес пошуку значень потрібних показників і процес створення звітів обов'язково використовує індекси. Повідомлення за різним типом розміщені у відповідних колекціях MongoDB. За тривалий за час експлуатації сховища накопилось до 64 індексів на колекцію. Ця межа є такою, що технічно не може бути перевищена і є блокуючим фактором у залученні нових постачальників та споживачів, для яких часто потрібно використовувати поля у запитах такі, які до цього не використовувалися

і відповідно індекси для таких полів відсутні [2]. Будь-який клієнт може надіслати запит і із ненульовою ймовірністю у запиті буде критерій відбору із поля, індексу якого не існує, без використання індексу, запит може тривати від 2 до 6 годин. Згідно політик підключення до сховища (**Connect timeout**) та очікування відповіді (**Socket timeout**) необхідність налаштувань таких багаточасових очікувань у споживачів є неприйнятним і зазвичай в умовах по-замовчуванню дорівнюють 30 секундам.

Очікуваний результат

В загальному клієнтів задовольнить сервіс, який відповідає за 5-10 секунд та дає змогу гнучко задавати довільні набори полів та їхні значення у запитах, також як важливе покращення, але не критичне – передбачити можливість швидкого пере налаштування запиту із зміною пріоритетів клієнта.

Аналіз джерел та останніх досліджень показав, що для досягнення високої доступності важливо зосередитися на ключових компонентах, таких як надійне апаратне забезпечення, мережеві рішення, програмне забезпечення, а також стратегії реплікації даних і автоматичного переключення при виникненні помилок (failover). Крім того, важливо враховувати і метрики, які допомагають оцінити ефективність реалізації високої доступності, зокрема uptime, RTO (Recovery Time Objective) та RPO (Recovery Point Objective) [1] (Рис.1).

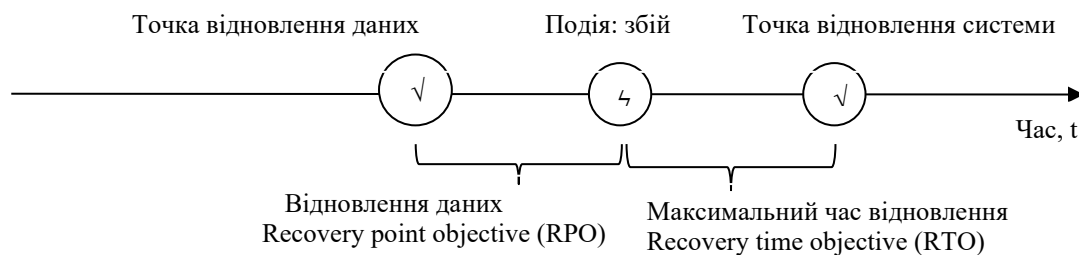


Рис.1. Розташування метрик відновлення у часі

Ці показники допомагають визначити, наскільки швидко система може бути відновлена після збою та яка кількість даних може бути втрачена без критичних наслідків для бізнесу.

Розглядаючи існуючі стратегії забезпечення високої доступності, ми бачимо широкий спектр підходів, від кластеризації серверів [3], що дозволяє розподіляти навантаження та забезпечувати відмовостійкість, до використання хмарних технологій [4, 5], які пропонують георозподіленість та еластичність ресурсів. Особливої уваги заслуговує реплікація даних, яка є ключовим елементом для забезпечення цілісності інформації [3].

Технології, які використовуються для досягнення цієї мети, різноманітні та постійно розвиваються. Від load balancing, який дозволяє ефективно розподіляти трафік між серверами, до систем аварійного переключення та розробки планів відновлення після аварій (Disaster Recovery plans), кожна технологія відіграє свою роль у побудові відмовостійкої системи [4].

Таким чином, глибоке розуміння принципів високої доступності, знання сучасних стратегій та технологій, а також аналіз реальних випадків їх застосування є ключовими для створення ефективних та надійних IT-інфраструктур.

Технології CI/CD

Зосередимося на огляді того, яку роль відіграють технології Continuous Integration (CI) та Continuous Delivery/Deployment (CD) у високій доступності [6, 7]. Ці методології є фундаментом для автоматизації процесів розробки та розгортання, спрямованих на підвищення якості продукту та ефективності робочих процесів.

Визначення CI/CD та їхня роль у високій доступності:

1. Continuous Integration (CI) — це практика, у якій розробники регулярно, часто кілька разів на день, інтегрують свій код у спільну гілку репозиторію [8]. Це дозволяє автоматично виконувати тести на кожну зміну коду, що значно знижує ймовірність появи помилок та спрощує виявлення та усунення проблем на ранніх етапах розробки.

2. Continuous Delivery є продовженням CI, яке забезпечує автоматичне розгортання всіх змін коду в тестове, проміжне або стейджингове середовище після успішного проходження тестів [9].

3. Continuous Deployment — ще одна стадія цього процесу, подальше розширення практики, що дозволяє автоматично розгортати зміни в production-середовище, за умови успішного проходження всіх тестів та перевірок [8, 9].

Застосування CI/CD має безпосередній вплив на підвищення високої доступності систем, адже дозволяє швидко вносити зміни, виявляти та усувати помилки, тим самим підтримуючи систему в актуальному та стабільному стані. Впровадження цих практик веде до зменшення часу простою та підвищення надійності служб.

Інструменти CI/CD

На ринку існує широкий вибір інструментів CI/CD, кожен з яких має свої унікальні переваги та недоліки. Найпопулярніші серед них:

- Jenkins: Відкритий інструмент, який пропонує величезну кількість плагінів та велику гнучкість у конфігурації процесів CI/CD [10, 11]. Однак, висока гнучкість може призвести до складностей у налаштуванні та управлінні.

- GitLab CI/CD: Вбудовані функції CI/CD в систему управління версіями GitLab, що дозволяє з легкістю налаштувати процеси автоматизації без необхідності встановлення зовнішніх інструментів [12].

- CircleCI: Хмарний сервіс, який пропонує швидку настройку та виконання процесів CI/CD з високою продуктивністю та масштабованістю [11].

- Travis CI: Хмарний сервіс, який інтегрується з GitHub, забезпечуючи простоту використання та налаштування [11].

Кожен з цих інструментів має свої особливості, які слід ретельно вивчати щоби обирати та застосовувати найкращий інструмент для конкретних проектів чи організацій.

Технології CI/CD є невід'ємною частиною сучасної розробки програмного забезпечення, спрямовані на підвищення якості продукту, ефективності процесів розробки та, зокрема, високої доступності систем. Вибір правильних інструментів CI/CD та їх адаптація під конкретні потреби проекту можуть значно вплинути на успішність впровадження цих практик.

Модель успішного розгалуження гілок в розробці

Робота технологій CI/CD базується на моделі розгалуження гілок у процесі розробки програмного забезпечення, та є ключовим елементом управління версіями та співпраці в команді розробників. Модель розгалуження впливає на ефективність розробки, можливість швидкого впровадження змін та, в кінцевому підсумку, на високу доступність продукту. Саме мітки станів системи є ключовою умовою коректної роботи технології CI/CD.

Основні поширені моделі розгалуження:

1. Модель Git Flow (автор Вінсент Дрійссен [12, 13]), є однією з найпопулярніших стратегій розгалуження, що передбачає використання фіксованого набору гілок для різних цілей — розробки, випусків, функціоналів та гарячих поправок. Ця модель дозволяє ефективно управляти релізами та паралельною розробкою нового функціоналу, забезпечуючи стабільність продукту та спрощення процесу розгортання.

2. Модель GitHub Flow. Вона є простішою альтернативою згаданій Git Flow [13, 14]. Модель GitHub Flow зосереджується на одній головній гілці (master/main), з якої створюються нові гілки для розробки функціоналу чи виправлення помилок. Зміни інтегруються назад у головну гілку через pull-requests, що забезпечує неперервне розгортання. Ця модель ідеально підходить для проектів, які мають на меті високу швидкість розробки та релізів.

3. Модель GitLab Flow комбінує елементи обох попередніх підходів, пропонуючи більшу гнучкість [13, 14]. Вона включає в себе використання гілки master для стабільної версії, окремих гілок для розробки функціоналу та додаткових механізмів для управління релізами через стейджингові та продакшн-середовища. GitLab Flow забезпечує баланс між стабільністю та швидкістю розробки, дозволяючи гнучко налаштувати процес під потреби проекту.

При виборі моделі розгалуження важливо враховувати кілька ключових факторів, серед яких: розмір та склад команди розробників, частота та складність релізів, а також загальні вимоги до стабільності та доступності продукту. Для проектів з високими вимогами до стабільності та регулярними, але добре контрольованими релізами, Git Flow може стати оптимальним вибором. Проекти, що прагнуть до швидкої розробки та випуску, можуть виграти від простоти та гнучкості GitHub Flow або GitLab Flow.

З урахуванням вимог до високої доступності, важливо обрати модель, що дозволяє швидко реагувати на помилки, ефективно управляти версіями та забезпечувати неперервний процес розробки та розгортання. Моделі розгалуження повинні сприяти автоматизації процесів, зниженню ризиків помилок та підвищенню якості кінцевого продукту.

У підсумку, вибір моделі розгалуження — це ключове рішення, що впливає на продуктивність команди, швидкість розробки та стабільність продукту. Врахування специфіки проекту та потреб бізнесу допоможе вибрати найбільш ефективний підхід до управління версіями та розгалуження.

Виклад основного матеріалу дослідження

Перерахуємо та проаналізуємо можливі стратегії та підходи до розробки та тестування програмного забезпечення, орієнтовані на забезпечення високої доступності.

1. Реорганізувати сховище, збільшення типів даних і створення нових колекцій та використати шардування — горизонтальне масштабування [5].

2. Збільшення продуктивності сховища — вертикальне масштабування [5].

3. Узагальнення запитів користувачів — пропозиція створювати тільки такі запити, як використовують наявні індекси [15].

4. Оптимізувати існуючі індекси — видалити непотрібні індекси, використовувати тільки узагальнені індекси [15].

5. Використати часткові індекси [2].

6. Переглянути архітектуру додатку, створити кешування, асинхронне оброблення, оптимізувати існуючі запити [1].

7. Обмежити доступ користувачів у випадку пікових навантажень – використати архітектурний патерн – Circuit Breaker [15].

8. Створити сервіс із новою архітектурою.

9. Використання монадних структур які будуть переміщувати сторонні ефекти в контекст [16].

Проаналізуємо запропоновані стратегії.

1. Реорганізація сховища змушує робити зміни у додатку, змінювати істоти (структури даних на рівні представлення, бізнес-логіки та у сховищі) як такі, що відображають структури (шаблони) які заповнюються даними. Обов'язкові зміни відбудуть у структурах які записуються в сховище у відповідну колекцію. Такі зміни викликають необхідність закладати додатковий час на внесення змін та тестування і часто є неприйнятними для поточних клієнтів із визначеним та вичерпаним бюджетом та уже доставленим і стабільним рішенням.

2. Збільшення продуктивності зазвичай призводить до експоненційного збільшення вартості обладнання при кратному збільшенні продуктивності. Що не є прийнятним оскільки очікується оптимізація, яка зменшить час очікування із декількох годин до одиниць секунд.

3. Якщо дозволяти користувачам створювати такі запити, які можуть використовувати наявні індекси, тоді має бути механізм забезпечення коректності дій користувачів. Користувач не є розробник сервісу чи фахівцем по сховищам даних. Ідея надати користувачу такі права, які призведуть до збоїв системи і відмову обслуговування інших користувачів не є прийнятною.

4. Якщо відстежувати частоту використання запитів і відповідні їм індекси, то можна вивільнити позиції для створення нових індексів. Але потрібна взаємодія із користувачами для яких створювалися такі індекси. Треба виявити усі можливі запити із небезпечними комбінаціями полів і заборонити їх. Така задача по трудомісткості є аналогічна внесенню змін у додаток, у загальному такий підхід також неприйнятний. Такий підхід показав гарні результати в умовах використання цього правила від початку взаємодії з клієнтами та постійного ретельного планування взаємодії із новими полями у структурі даних.

5. Використати часткові індекси — ідея гарна і легко втілюється, якщо виявити майбутню проблему дефіциту вільних місць для індексів. Але не вирішується сама проблема відправлення запиту із відсутнім індексом для обслуговування такого запиту. Тобто такий підхід є прийнятним але якщо контролювати формат запиту — дозволяти тільки вивірені комбінації параметрів. На запити перекладається частина бізнес-логіки у випадку запитів із декількох колекцій. Таке рішення ламає архітектуру сервісу.

6. Гарний підхід для підвищення швидкості оброблення, але доволі часто будуть зустрічатися кейси, коли користувач викриє таку комбінацію параметрів, що буде відмова в обслуговуванні, результат не буде створений, тому кешування не відбудеться, інші користувачі отримають помилку — відмовлення в обслуговуванні. Такий підхід не є прийнятним за виключенням організації асинхронного оброблення. Такий функціонал можна внести в архітектуру сервісу і використовувати. Але враховуючи об'єми даних та частоту їх оновлення варіант не є доцільним для використання.

7. Таке рішення дозволить відключати клієнтів із довгими запитами або переривати запити для звичайних користувачів, коли обслуговується клієнт із «важким» запитом. Рішення вносить деякі покращення у взаємодію клієнтів із сервісом – якщо система не готова обслуговувати, то принаймні не буде перевантажуватися сховище, а клієнт одразу отримає відповідь по відмову в обслуговуванні. В цілому таке рішення не вирішує проблему і часто використовується у мікросервісній архітектурі, зберігаючи систему цілому від перевантаження та зупинки обслуговування. Рішення не відповідає очікуванням, оскільки залишаться клієнти, які будуть відправляти «важкі» запити, понаднормово очікувати та в цілому блокувати роботу сервісу.

8. Добре рішення, але негативним є необхідність значних витрат часу на створення та тестування нового сервісу та додаткові фінансові витрати на розробку та онбординг існуючих клієнтів.

9. Теорія категорій пропонує винести сторонні ефекти у контекст та зосередитися на бізнес-логіці. На практиці таке не внесе поліпшення у зменшення часу відповіді чи підвищить доступність сервісу, оскільки такий рефакторинг відбувається без врахування роботи сховища. Не очікується змін у взаємодії із сховищем лише поліпшується загальна архітектура сервісу – розробники мають можливість модифікувати безпосередньо бізнес-логіку не відволікаючись на оброку сторонніх ефектів. Також слід відмітити, що підтримка монад них структур вимагає наявності функції як типа даних у мові програмування, на якій створюється сервіс.

Вибір рішення

Після проведення аналізу виявлено, що жодний із варіантів не дає відповіді на запитання як ґрунтуючись на досягненнях у дисципліні комп'ютерних технологій зменшити час обслуговування на 3-4 порядки без критичних змін у архітектурі сервісу та в умовах забезпечення стабільного обслуговування існуючих клієнтів. Тому рішенням може бути гібридним – часткове залучення різних підходів. І таке можливо, шляхом впровадження додаткової активності у поточну модель розгалуження – Git Flow, а саме створення дозволів для клієнтів використовувати набори параметрів для заданих типів даних. Такий підхід дозволяє долучити обслуговування нових клієнтів із новими запитами але використання таких запитів супроводжується повноцінною процедурою створенням та узгодження запиту із клієнтом, тестуванням та покроковим рухом рішення через усі середовища розробки до роботи у фінальному середовищі – production.

Таке рішення не має різких негативних недоліків окрім часу на додаткову розробку запитів та необхідність клієнтам слідувати встановленим контрактам взаємодії із сервісом.

Отже, представлено новаторську пропозицію, яка втілює інтеграцію практик Continuous Integration/Continuous Deployment (CI/CD), оптимізованих моделей розгалуження гілок та покрокової доставки рішення з розробницького середовища у production. Цей підхід спрямований на підвищення ефективності процесів розробки та тестування, а також на забезпечення високої доступності рішень, доставлених замовнику. Наслідком такого рішення стало зменшення часу відгуку відповідно до умов контракту із замовником та виключення помилкових ситуацій пов'язаних із часом очікування відповіді.

Впровадження описаної стратегії відкриває новий підхід для вирішення аналогічних класів задач пов'язаних із типом взаємодії з користувачами, коли користувацькі налаштування є обов'язковою частиною сервісу і гарантовано несуть загрозу недоступності сервісу. Цей підхід відкриває нові можливості для досягнення високої доступності систем, зниження ризиків та підвищення задоволеності користувачів.

Узагальнення

Оглянемо ключові аспекти дослідження. Наше дослідження зосередилося на інтеграції передових практик Continuous Integration/Continuous Deployment (CI/CD) та оптимізованих моделей розгалуження гілок з метою створення ефективної стратегії розробки та тестування. Було продемонстровано, як підхід може значно підвищити швидкість розробки, якість продукції та забезпечити надійну доступність сервісів для кінцевих користувачів.

Слід відзначити, що протягом практичної реалізації стратегії наштовхнулися з низкою викликів, серед яких було важливо відзначити:

- Інтеграція нових процесів у існуючі робочі потоки, що вимагає часу на адаптацію та навчання команди.

- Вибір та налаштування інструментів CI/CD, що найкраще відповідають потребам проекту.

- Оптимізація моделей розгалуження для забезпечення гнучкості та ефективності процесу розробки.

Додатковими рішеннями підвищенню високої доступності також сприяють організаційні фактори і стратегії, зокрема:

- Проведення тренінгів та семінарів для команди розробників з метою підвищення їх компетенцій в області CI/CD та роботи з новими інструментами.

- Поступове впровадження змін, що дозволяє команді адаптуватися без різких переходів від звичних методів до нових практик.

- Тісна співпраця з замовниками для забезпечення прозорості процесу розробки та впровадження змін, що відповідають їхнім вимогам та очікуванням.

Узагальнюючи дослідження, слід підкреслити важливість інтеграції сучасних технологій та методологій у процесі розробки та тестування для досягнення високої продуктивності, якості та доступності програмних продуктів. Впровадження запропонованої стратегії демонструє значний потенціал для підвищення ефективності розробки, зменшення часу виходу на ринок та підвищення задоволеності користувачів.

Підсумки дослідження

На основі аналізу запропонованих підходів була розроблена новітня стратегія багатокрокового доставляння сервісу користувачам, здобуто такий результат як допустимий час очікування відповіді до 5 секунд. Впровадження цієї стратегії посприяло досягненню доступності програмних продуктів – 99,999%, забезпечивши надійність та стабільність сервісів для кінцевих користувачів.

Рекомендації для подальших досліджень

На основі отриманих результатів рекомендується подальші дослідження в наступних напрямках:

1. Глибше дослідження індивідуальних компонентів CI/CD та моделей розгалуження, щоб визначити їх найбільш ефективні конфігурації для різних типів проектів та робочих середовищ.

2. Аналіз впливу організаційної культури та структури на успішність впровадження інтегрованих підходів до розробки та тестування, а також розробка методик їх оптимізації.

3. Розробка та тестування нових інструментів та технологій, які можуть допомогти в подальшому покращенні процесів CI/CD та розгалуження гілок, особливо у контексті використання штучного інтелекту та машинного навчання для автоматизації тестування та виправлення помилок.

Запропоноване рішення пройшло апробацію у Citibank, підрозділу Citigroup, успішно впроваджене у бізнес-продукт компанії, виведено у production та надихнуло відобразити як науковий результат.

References

1. Niall Richard Murphy, Betsy Beyer, Chris Jones, and Jennifer Petoff. Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media, 2016, 550 p.
2. Kristina Chodorow, Shannon Bradshaw, Eoin Brazil. MongoDB: The Definitive Guide, 3rd Edition. O'Reilly Media, Incorporated, 2018, 425 p.
3. David Kopec, Classic Computer Science Problems in Python. Manning Publications; 1st edition. 2019, 224 p.
4. Sam Newman. Building Microservices: Designing Fine-Grained Systems, 2nd edition. O'Reilly Media, 2021, 612 p.
5. Kris Jamsa, Cloud Computing. Jones & Bartlett Learning, 2022, 400 p.

6. Paul M. Duvall, Steve Matyas, Andrew Glover, Continuous Integration: Improving Software Quality and Reducing Risk by. Addison-Wesley Professional, 2007, 336 p.
7. Joakim Verona, Practical DevOps. Packt Pub Ltd., 2016, 217с.
8. Jez Humble and David Farley. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional, 2012, 512 p.
9. Emily Freeman. DevOps for Dummies. For Dummies, 2019, 368 p.
10. Nicole Forsgren, Jez Humble, and Gene Kim. Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press, 2018, 288 p.
11. Mitesh Soni. Implementing DevOps with Microsoft Azure. Packt Publishing, 2017, 422 p.
12. Scott Chacon and Ben Straub. Pro Git. Apress, 2014, 458 p.
13. Vincent Driessen. A successful Git branching model. [Електронний ресурс]. – Режим доступу: <https://nvie.com/posts/a-successful-git-branching-model>. – Назва з екрану. 05.01.2010
14. Gene Kim, Kevin Behr, and George Spafford. The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win IT Revolution Press, 2013, 345 p.
15. Martin Fowlerю Circuit Breaker. [Електронний ресурс]. – Режим доступу: <https://martinfowler.com/bliki/CircuitBreaker.html>. – Назва з екрану. 06.03.2014
16. Saunders Mac Lane. Categories for the Working Mathematician. Springer Science & Business Media, 1998, 314 p.