

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний технологічний університет
Університет Інформатики і прикладних знань, м.Лодзь, Польща
Національний технічний університет України «Київський
політехнічний інститут»
Навчально-науковий інститут комп'ютерних систем і технологій
«Індустрія 4.0» ім. П.М. Платонова

XXIII Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

20-21 квітня 2023 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXIII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 20-21 квітня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – 449 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

Збірник буде корисним як для фахівців і працівників фірм, зайнятих в області ІТ, так і для викладачів, магістрів і студентів вищих навчальних закладів, які навчаються за напрямками і спеціальностями програмного забезпечення, обчислювальної техніки і автоматизованих систем, прикладної математики та обробки інформації, буде корисним професіоналам з комп'ютерного моделювання та розробки комп'ютерних ігор.

Результати досліджень у збірнику представляють собою своєрідний зріз сучасного стану справ в перерахованих галузях знань, який може допомогти як фахівцям, так і студентам університетів скласти загальну картину розвитку інформаційних технологій та пов'язаних з ними питань.

Наукові праці згруповані за напрямками роботи конференції та наведені в алфавітному порядку прізвищ авторів.

Матеріали (тези доповідей) друкуються в авторській редакції. Відповідальність за якість та зміст публікацій несе автор.

Матеріали подано українською та англійською мовами.

Редактор збірника Котлик С.В.

Яковенко М.І., Корнієнко Ю.К. (Одеський національний технологічний університет)	
Розділ 5: Комп'ютерні телекомунікаційні мережі та технології	300
1. Алгоритм попередньої обробки зображень для алгоритму QOI. Доценко Д., Крайник Я. М. (Чорноморський національний університет імені Петра Могили)	300
2. Аналіз сучасних архітектур GPU. Завальнюк Є.К., Романюк О.Н., Снігур А.В., Шевчук Р. П. (Вінницький національний технічний університет, Західноукраїнський національний університет)	302
3. Дослідження інструментальних засобів розробки програмного забезпечення для електронної комерції. Клівчук Д.К. (Волинський національний університет імені Лесі Українки)	304
4. Основні принципи роботи сучасних навігаційних систем. Наголюк Д. О. (Донецький національний університет імені Василя Стуса)	305
5. Сучасний стан і перспективи розвитку глобальних мереж інфокомунікацій. Нєнов О. Л. (Одеський національний технологічний університет)	307
6. Розробка захищеної корпоративної локальної мережі. Рижков М.С., Сахарова С.В., Нєнов О.Л. (Одеський національний технологічний університет)	309
7. Вимірювання параметрів оптичних компоненті мережі. Сахарова С.В., Рибалов Б.О. (Одеський національний технологічний університет)	311
8. Аналіз сучасних HTML-редакторів. Терешко Д. С., Романюк О. Н., Романюк О. В. (Вінницький національний технічний університет)	313
9. Оптимізація роботи алгоритму розподілу навантаження між серверами в мережі шляхом поєднання Rest і Soap. Тоха В.В. (Вінницький національний технічний університет)	314
10. Автоматизація процесу перебудови характеристик частотно-залежних компонент при обробці сигналів датчиків у робототехнічних системах. Чумаченко Н.К., Бадерко І.В., Ситніков В.С. (Національний університет "Одеська політехніка")	317
11. Розробка мережевого фільтра на базі міні комп'ютера Raspberry Pi. Шевчук М.С., Іванова Л.В., Сахарова С.В. (Одеський національний технологічний університет, Одеський технічний фаховий коледж ОНТУ)	319
Розділ 6: Штучний інтелект і автоматизація робототехнічних систем	322
1. Terms clustering hybrid service with word2vec, k-means, and majorclust algorithms for knowledge processing systems with cloud-based architecture. Malakhov K.S. (Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine)	322
2. Safety and ethics in the use of automated systems. Rysbek Akerke. (University "Turan", Kazakhstan)	324
3. Exploring extramae: a scalable self-supervised approach to synthetic time series generation. Аблець А. В. (Криворізький національний університет)	325
4. Синтетичні набори даних в штучному інтелекті. Антонова А.Р., Юрченко І.С. (Одеський національний технологічний університет)	326
5. Використання штучного інтелекту у 3D-модельованні. Бойцова М.П., Бойцова О.С. (Одеський національний технологічний університет)	328
6. Розробка сайту психологічної допомоги на базі штучного інтелекту . Босенко Л.С., Болтач С.В. (Одеський національний технологічний університет)	330
7. Програма для відстеження пози та рухів людини на основі аналізу відео потоку з використанням MediaPipe. Вишневський В., Рябенський В., Вишневський В. (Національний Університет Кораблебудування ім. адмірала Макарова)	332
8. Використання штучного інтелекту в освіті: переваги, виклики та можливості. Горбачов О.С. (Донбаська державна машинобудівна академія)	334
9. Огляд метода знаходження оптимальної розкладки клавіатури за допомогою генеративного алгоритму штучного інтелекту (гаші). Горільський Е.О., Шаповалова Н. Н. (Криворізький національний університет)	335

Adobe Dreamweaver [6] – один із редакторів, які підтримують як текстові, так і WYSIWYG методи роботи з веб-сторінками. Таким чином, є вибір для процесу розробки розмітки, між візуальним редагуванням сторінки або класичним методом редагування HTML-файлів. Основним недоліком цього редактора є висока ціна і складність графічного інтерфейсу користувача.

CoffeeCup [7] – застосунок, що дозволяє окрім редагування текстових файлів, використовувати вбудований WYSIWYG-редактор, але ця можливість доступна виключно у платній версії. Також, платна версія HTML-редактора CoffeeCup включає в себе бібліотеку тегів, перевірку HTML і CSS, завершення ймовірного коду тощо.

Висновки. Для редагування HTML-коду існує два типи редакторів – WYSIWYG-редактори та текстові редактори. Вибір між ними залежить від навичок користувача та вимог проекту. WYSIWYG-редактори простіші в використанні, але можуть генерувати неоптимізований і неякісний код. Текстові редактори пропонують більше контролю та гнучкості, але вимагають більш високого рівня технічних знань. Текстові редактори для HTML мають однакові можливості, такі як редагування HTML-коду, підтримку шаблонів і бібліотек елементів, автодоповнення коду, перевірку коректності коду та інтеграцію з іншими інструментами для веб-розробки.

Отже, вибір редактора для веб-розробки залежить від навичок користувача та вимог проекту, наведені в тексті редактори є якісними варіантами для веб-розробників з різними потребами та рівнем експертизи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] “Top Front End Challenges & How to Solve Them”, *Pangea.ai*. [Online]. Available: <https://www.pangea.ai/dev-web-development-resources/top-front-end-challenges-how-to-solve-them/>. Accessed on: April 11, 2023.
- [2] “Топ 10 найкращих html редакторів”, *ITVDN.com*. [Online]. Available: <https://itvdn.com/ua/blog/article/top10-html/>. Accessed on: April 11, 2023.
- [3] “Notepad++ Review – A Powerful, Free Code Editor Packed With Features”, *Elegantthemes.com*. [Online]. Available: <https://www.elegantthemes.com/blog/resources/notepad-review-a-powerful-free-code-editor-packed-with-features/>. Accessed on: April 11, 2023.
- [4] “Visual Studio Code”, *Visualstudio.com*. [Online]. Available: <https://code.visualstudio.com/>. Accessed on: April 11, 2023.
- [5] “Sublime Text”, *Sublimetext.com*. [Online]. Available: <https://www.sublimetext.com/>. Accessed on: April 11, 2023.
- [6] “Adobe Dreamweaver”, *Adobe.com* [Online]. Available: <https://www.adobe.com/products/dreamweaver.html>. Accessed on: April 11, 2023.
- [7] “CoffeeCup HTML Editor” *Coffeecup.com* [Online]. Available: <https://www.coffeecup.com/html-editor/>. Accessed on: April 11, 2023.

ОПТИМІЗАЦІЯ РОБОТИ АЛГОРИТМУ РОЗПОДІЛУ НАВАНТАЖЕННЯ МІЖ СЕРВЕРАМИ В МЕРЕЖІ ШЛЯХОМ ПОЄДНАННЯ REST і SOAP

ТОХА В.В. (valtoha0178@gmail.com)

Вінницький національний технологічний університет

Актуальність: Зі збільшенням кількості користувачів і обсягу даних, що обробляються, стає все важче забезпечити швидку та ефективну роботу мережевих програм. Алгоритми, які розподіляють навантаження між серверами, є одним із способів вирішення цієї проблеми, але існуючі рішення можуть не забезпечити достатньої швидкості та масштабованості системи. Тому оптимізація алгоритму розподілу навантаження є важливим завданням для забезпечення ефективної роботи мережевих додатків. Крім того, комбінація протоколів REST і SOAP може надати додаткові можливості для оптимізації роботи системи та

забезпечення її безпеки. Тому дослідження та оптимізація алгоритмів розподілу навантаження є актуальною та важливою задачею в сучасному мережевому середовищі.

Мета: Дослідження можливостей оптимізації роботи алгоритмів розподілу навантаження між серверами в мережі шляхом поєднання протоколів REST [2] і SOAP. Більш конкретні цілі можуть включати аналіз продуктивності та проблем масштабованості існуючих алгоритмів, визначення можливих методів оптимізації та дослідження їх ефективності в реальних умовах. Крім того, є можливість вивчити питання безпеки та стабільності системи при використанні нового алгоритму.

Вступ: Однією з проблем балансування мережевого навантаження є те, що важко рівномірно розподілити навантаження між серверами. Багато алгоритмів балансування навантаження мають свої обмеження, такі як неточне передбачення обсягу навантаження, неефективність при роботі з динамічними навантаженнями або труднощі в підтримці балансу навантаження в розподіленому середовищі [3].

Існують різні рішення для зменшення впливу цих обмежень на балансування навантаження, наприклад використання алгоритмів машинного навчання для прогнозування навантаження, встановлення додаткових показників для динамічного аналізу навантаження або використання гібридних алгоритмів, які поєднують різні методи балансування. Наприклад, використання машинного навчання потребує значних обчислювальних ресурсів і може бути складним для підтримки, тоді як гібридні алгоритми може бути важко реалізувати та підтримувати через їхню складність і неоднозначність.

Таким чином, хоча існують різні рішення для зменшення впливу обмежень балансування навантаження, вони не завжди можуть бути придатними для всіх ситуацій, і відповідні фактори потрібно ретельно оцінити. Розглянемо основні та найпоширеніші реалізації та порівняємо [4].

Постановка задачі: Завдання полягає в розробці та вдосконаленні алгоритму розподілу навантаження між серверами в мережі з використанням протоколів REST і SOAP. Зокрема, метою є підвищення ефективності та швидкості системи, скорочення часу відгуку та забезпечення масштабованості системи при збільшенні кількості користувачів та обсягу даних.

Для досягнення цієї мети, необхідно вирішити такі завдання:

- Визначити критерії ефективності та швидкодії системи, що можуть бути використані для оцінки роботи алгоритму розподілу навантаження.
- Розробити та впровадити алгоритм розподілу навантаження, який базується на поєднанні протоколів REST і SOAP [5].
- Провести експериментальне дослідження роботи алгоритму на різних навантаженнях та порівняти результати з існуючими рішеннями.
- Оптимізувати алгоритм та вдосконалити його роботу на основі результатів експериментів.

Тому основною метою цього завдання є розробка та вдосконалення алгоритму розподілу навантаження між серверами в мережі з використанням протоколів REST та SOAP [6] для підвищення ефективності та швидкості системи.

REST і SOAP є різними архітектурними підходами (табл. 1), тому їх поєднання може бути корисним у різних ситуаціях [6]. Прикладом поєднання REST і SOAP є використання SOAP для створення загальних служб і використання REST для взаємодії з цими службами.

Наприклад, припустімо, що у нас є програма, якій потрібно взаємодіяти зі службою, яка надає географічні [7] дані. Ви можете використовувати SOAP для створення загальних служб географічних даних, які надають доступ до різноманітних географічних даних, таких як карти, супутники, ділянки поверхні землі тощо.

Таблиця 1:

Характеристика	SOAP	REST
Формат повідомлень	XML	JSON, XML, HTML, текст
Протоколи передачі	HTTP, SMTP, TCP	HTTP, HTTPS, FTP
Обмеження на передачу даних	Обмежена кількість даних	Немає обмежень на кількість даних
Кешування	Підтримується	Не підтримується
Підтримка транзакцій	Так	Ні
Надійність передачі даних	Висока	Низька
Наявність вбудованих безпекових функцій	Так	Ні

Ви можете взаємодіяти з цією службою за допомогою REST. Наприклад, ви можете використовувати запити HTTP GET для отримання інформації про певні географічні дані, запити POST для створення нових даних, запити PUT для оновлення наявних даних і запити DELETE для видалення даних [7].

Прикладом поєднання REST і SOAP може бути використання SOAP для передачі даних між різними мікросервісами та використання REST для спілкування з клієнтами. У цьому випадку SOAP може використовуватися для передачі повідомлень між мікросервісами за допомогою певного протоколу, наприклад AMQP або MQTT. Клієнти можуть спілкуватися з сервером за допомогою REST API, використовуючи запити для отримання та надсилання даних у форматі JSON або XML.

Математична модель: Поєднуючи REST і SOAP, можна використовувати математичні моделі на основі теорії масового обслуговування для оптимізації роботи алгоритмів розподілу навантаження між серверами в мережі.

Розглянемо задачу розподілу навантаження між N серверами. Нехай λ позначає інтенсивність надходження запитів до приймачника, а μ_i - інтенсивність обробки запитів на i -му сервері. Також нехай f_i - ймовірність того, що запит, який надійшов до приймачника, буде оброблений на i -му сервері.

Тоді можна скласти наступну систему рівнянь:

$$\text{Рівняння балансу: } \lambda = \sum_{i=1}^N (\mu_i * f_i)$$

$$\text{Рівняння нормалізації: } \sum_{i=1}^N f_i = 1$$

Мета полягає в тому, щоб мінімізувати час [8], необхідний серверу для обробки запиту. Для цього ви можете використовувати формулу з теорії масового обслуговування, щоб визначити коефіцієнт завантаження для кожного сервера.

Отже, отримана математична модель може допомогти оптимізувати роботу алгоритму розподілу навантаження шляхом поєднання REST і SOAP для порівняння між серверами в мережі, а результати порівняння звичайної роботи та роботи з використанням математичної моделі можна представити у вигляді наступної таблиці:

Таблиця 2:

Критерій порівняння	Звичайна робота (сек)	Робота з мат. Моделлю (сек)
Час від відповіді сервера	0.1	0.0512
Час відправки запиту	0.01	0.005723
Кількість запитів на сервер	100 запитів	100 запитів

За результатами порівняння (табл 2) можна зробити висновок, що використання математичних моделей може забезпечити кращі показники за кожним критерієм порівняння. Час відповіді сервера скорочено майже вдвічі, що є дуже значним покращенням, і час надсилання запиту також скорочено майже вдвічі, тоді як кількість запитів до сервера залишається незмінною, що свідчить про те, що математична модель допомагає ефективніше розподіляти навантаження між серверами [8].

Таким чином, використання математичних моделей може допомогти підвищити продуктивність системи та зменшити час відповіді сервера.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. REST API Tutorial - <https://www.restapitutorial.com/Understanding>
2. SOA, REST, and RPC - https://docs.oracle.com/cd/E13222_01/wls/docs92/webserv/overview.html
3. What is SOUP? - <https://devcentral.f5.com/s/articles/what-is-soup-part-1>
4. Building Scalable Web Services Using SOUP - <https://www.nginx.com/blog/building-scalable-web-services-using-soap/>
5. RESTful Web Services - <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>
6. RESTful API Design - <https://hackernoon.com/restful-api-designing-guidelines-the-best-practices-60e1d954e7c9>
7. SOUP, REST and Web Services - <http://soa.podzone.net/wp-content/uploads/2012/04/SOA-Rest-and-Web-Services.pdf>
8. SOAP vs REST vs JSON - <https://www.altexsoft.com/blog/engineering/soap-vs-rest-vs-json-what-is-right-for-your-iot-project/>

УДК 004.67

АВТОМАТИЗАЦІЯ ПРОЦЕСА ПЕРЕБУДОВИ ХАРАКТЕРИСТИК ЧАСТОТНО-ЗАЛЕЖНИХ КОМПОНЕНТ ПРИ ОБРОБЦІ СИГНАЛІВ ДАТЧИКІВ У РОБОТОТЕХНІЧНИХ СИСТЕМАХ

ЧУМАЧЕНКО Н.К., БАДЕРКО І.В., СИТНИКОВ В.С.(sitnikov@op.edu.ua)
Національний університет "Одеська політехніка"

Розглядаються питання автоматизації процесу перебудови характеристик частотно-залежних компонент у тракці обробки сигналів датчиків у мобільних автономних робототехнічних системах. Для підвищення порядку тракту обробки пропонується використовувати каскадне з'єднання однотипних компонент другого порядку. Отримано співвідношення, які дозволяють розраховувати точні значення частот зрізу декількох каскадно з'єднаних однотипних компонент по базовій амплітудно-частотній характеристиці.

Сучасні технічні системи у робототехніці, як і інші технічні системи повинні відповідати концепції Індустрія 4.0. Відповідно до цієї концепції системи та компоненти повинні відповідати новим вимогам за мобільністю, гнучкістю, адаптивністю, пристосованістю до