

ДОДАТОК А

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

21. Kiriу V.V, Sheiko, I., Petrova, R., Optimization of management information support as a basis for organizational transformations at an enterprise, Periodicals of Engineering and Natural Sciences. 2019. V.7. N. 2. P. 679-689

22. Кравець Н. С., Інтерактивна імітація та аналіз розкрашеної сітки Петрі з використанням алгебри предикатних операцій, Праці п'ятої всеукраїнської міжнародної конференції 27 листопада -1 грудня 2000р., Україна, Київ. С.313-316.

23. Shubin I.Yu, A. Kozyriev, V. Liashik, G. Chetverykov, Methods of Adaptive Knowledge Testing Based on the Theory of Logical Networks, Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), Kharkiv, Ukraine, April 23-24, 2021. – Volume I, P. 1184-1193. (CEUR, SCOPUS)

24. K. Smelyakov, A. Chupryna, o. Bohomolov and N. Hunko, “The Neural Network Models Effectiveness for Face Detection and Face Recognition”, 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Нечволод Вадим Юрійович каф. ПІ

ID перевірки:
1016288815

Дата перевірки:
28.05.2024 07:35:28 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
28.05.2024 09:10:15 EEST

ID користувача:
94949

Назва документа: 2023_М_ПІ_ІПЗм_22_5_Гавриш_Д_Л_скорочений

Кількість сторінок: 60 Кількість слів: 11306 Кількість символів: 84381 Розмір файлу: 1.11 MB ID файлу: 1016082317

18.3% Схожість

Найбільша схожість: 6.89% з Інтернет-джерелом (<https://ela.kpi.ua/handle/123456789/45337>)

18.1% Джерела з Інтернету 198 Сторінка 62

7.77% Джерела з Бібліотеки 32 Сторінка 66

1.57% Цитат

Цитати 2 Сторінка 67

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 28

ДОДАТОК В

Матеріали XXVIII міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» 16–18 квітня 2024 р. Том 6. Частина I. Конференція «Інформаційні інтелектуальні системи» на тему «Дослідження моделі побудови мікросервісної архітектури за допомогою безсерверних обчислень»

УДК 004.4

DOI: <https://doi.org/10.30837/IYF.IIS.2024.332>**ДОСЛІДЖЕННЯ МОДЕЛІ ПОБУДОВИ МІКРОСЕРВІСНОЇ
АРХІТЕКТУРИ ЗА ДОПОМОГОЮ БЕЗСЕРВЕРНИХ ОБЧИСЛЕНЬ**

Гавриш Д. Л., Скібін О. О.

Науковий керівник – д.т.н., проф. Смеляков К. С.

Харківський національний університет радіоелектроніки, каф. ПІ

м. Харків, Україна

e-mail: dmytro.havrysh@nure.ua

This thesis proposes an intelligent cross-border logistics system using IoT technology to enhance efficiency and security. Traditional systems face challenges like high computing loads and security risks. To overcome this, the paper suggests a platform with serverless architecture and microservices, distributing services across logistics companies. It employs API gateways for security and asymmetric encryption for data security. Experiments show reduced costs and complexity, improved security, throughput, and latency in cross-border transactions.

У зв'язку з поступовим розвитком глобальної економіки та загостренням процесу глобалізації, міжнародна торгівля стала невід'ємною складовою сучасної економіки. Логістика, що відіграє ключову роль у сприянні міжнародної торгівлі, постійно прогресує та інновується, сприяючи розвитку інтелектуалізації та цифровізації міжнародної логістики. Поява технології "Інтернет речей" (IoT) призвела до поширення міжз'язаних пристроїв та систем, що охоплюють широкий спектр сенсорів, розумних пристроїв та систем управління логістикою, задіяних у міжнародних логістичних операціях. Значна кількість даних, що генерується цими пристроями та системами, дозволяє ефективну взаємодію та обмін інформацією через Інтернет, що значно підсилює прогрес інтелектуалізації міжнародної логістики.

З поширенням інтеграції технологій питання безпеки IoT отримало підвищену увагу. Взаємопов'язаність пристроїв та систем у межах IoT потребує підключення до Інтернету, за допомогою якого запити з розподілених пристроїв централізовано обробляються та відповідаються. Однак такий централізований підхід має в собі вроджені вразливості безпеки, включаючи мережеві атаки, витоки даних, порушення конфіденційності та схильність до порушень безпеки системи. Особливо в області міжнародної логістики інформаційна безпека та захист стають ще більш критичними та складними через участь міжнародної торгівлі, митниць та факторів транспортування.

Децентралізовані характеристики сценаріїв IoT ефективно гармонізуються з розподіленою архітектурою застосунків. Два провідні парадигми проектування архітектури програмного забезпечення, а саме мікросервісна архітектура та "подієво-орієнтованих" архітектур, в якому є безсерверні обчислення, здобули широке використання в різних галузях як

невід'ємні складові розподіленої архітектури застосунків. Мікросервісна архітектура сприяє побудові високо гнучких та масштабованих систем, перевершуючи складні централізовані методи обробки з точки зору масштабованості, підтримуваності та розширюваності. Відповідно, вона пропонує вищу адаптабельність до змін бізнес-вимог, покращуючи надійність та продуктивність системи. З іншого боку, архітектура яка використовує безсерверні обчислення пропонує більш гнучкий та універсальний підхід до розробки, що прискорює впровадження нових застосунків, при цьому зменшуючи витрати на розробку та експлуатацію.

Поява безсерверних обчислень, як нового шаблону розгортання програмного забезпечення, приносить значні переваги, такі як операції без обслуговування та моделі вартості "плати-лише-за-використання". Тим часом, мікросервісна архітектура, відмінна від традиційної монолітної архітектури програмного забезпечення, з'явилася для вирішення вимог сучасних інтернет-сервісів, включаючи аспекти високої конкурентоспроможності, високої продуктивності та високої доступності [1]. Вона має значну економічну цінність [2]. Хоча мікросервісна архітектура набули популярності як парадигми розробки [3] розподілених архітектур програмного забезпечення, виникають виклики та складнощі при їх застосуванні в контексті безпеки IoT міжнародної логістики. Один із значних викликів полягає в забезпеченні безпеки та конфіденційності комунікації між мікросервісами та функціями. Оскільки кожна служба в мікросервісній архітектурі має незалежний характер, різні вимоги безпеки та обмеження приватності можуть застосовуватися до різних служб. У цьому відношенні встановлення розумних контролів доступу та дозволів на дані для кожного мікросервісу та функції є надзвичайно важливим для запобігання несанкціонованому доступу та витоку інформації. Крім того, служби та функції в мікросервісній та "подієво-орієнтованих" архітектурі піддаються різноманітним загрозам безпеки, включаючи атаки типу "відмова в обслуговуванні" та атаки на внесення змін у інформацію. В області міжнародної логістики ці атаки можуть мати руйнівні наслідки, такі як втрата інформації.

Під час процесу проектування архітектури виникла складність у оцінці обсягу трафіку системи через введення нового бізнес-сценарію. Розгортання додатків безпосередньо на хмарних серверах за допомогою традиційних підходів потребувало би ручного горизонтального масштабування для адаптації до недостатнього трафіку, що може призвести до втрати ресурсів, якщо заздалегідь буде надано декілька серверів. Для вирішення цієї проблеми та відповідно до вимог бізнесу, була прийнята мікросервісна архітектура з безсерверними обчисленнями на основі AWS для реалізації. Шляхом розгортання мікросервісів на AWS Lambda та використання розподіленого обчислення, додаток може автоматично масштабуватися еластично в залежності від обсягу трафіку, а

витрати розраховуються відповідно до кількості запитів та часу обчислення. Для оперативної відповіді на змінні вимоги користувачів, підвищення ефективності розробки та прискорення поставки системи важливим є впровадження процесів безперервної інтеграції та безперервної доставки (CI/CD) для автоматизації всього процесу випуску. Автоматизація завдань, таких як компіляція коду, побудова та розгортання, мінімізує необхідність у ручному контролі кожної зміни та зменшує ризики системи шляхом превентивного виявлення потенційних проблем. Цей процес автоматизації охоплює не лише кодові дії, а й динамічне створення, знищення та оновлення ресурсів, необхідних для роботи системи, включаючи сервери та бази даних. Збирати та зіставляти інформацію з усіх ресурсів AWS, програм та служб, що працюють на aws та локально допомагає сервіс моніторингу CloudWatch [4], Lambda Insights надає автоматичні інформаційні панелі в консолі CloudWatch. На рис 1 наведено архітектуру для моніторингу за допомогою CloudWatch.

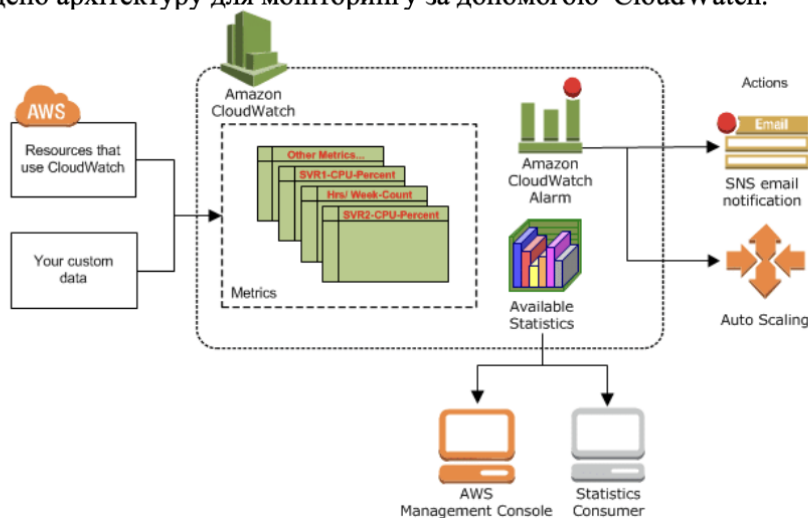


Рисунок 1 – Схема використання сервісу CloudWatch

Цей додаток спрямований на полегшення та оптимізацію процесів у сфері міжнародної логістики. Він забезпечує автоматизацію обробки даних, координацію та візуалізацію інформації про логістичні операції, що дозволяє зменшити час і зусилля, затрачені на вирішення рутинних завдань. Додаток також забезпечує безпечний обмін даними між учасниками логістичних процесів, забезпечуючи конфіденційність та цілісність інформації. Крім того, він надає можливість аналізу даних та надає рекомендації для оптимізації процесів, що дозволяє підприємствам зменшити витрати та підвищити ефективність своєї діяльності.

Оцінка якості моделі базується на конкретних критеріях, які включають:

1. Цілісність: Модель повинна враховувати всі аспекти міжнародної логістики, включаючи транспорт, митні процедури, складське господарство та інші, щоб забезпечити повноту та реалістичність аналізу.

2. Доступність: Важливо, щоб модель була легко доступною для користувачів у будь-який час і з будь-якого місця, щоб забезпечити зручність та ефективність використання.

3. Швидкодія: Модель повинна працювати швидко та ефективно, забезпечуючи миттєвий доступ до результатів аналізу та інформації для прийняття рішень.

4. Точність: Важливо, щоб модель надавала точні результати аналізу та прогнозування, щоб користувачі могли робити інформовані рішення.

Ці критерії допомагають забезпечити, що модель відповідає вимогам користувачів та сприяє покращенню ефективності логістичних процесів.

Безсерверні та мікросервіси є архітектурно пов'язаними технологіями, які використовують різні методи. Безсерверні та мікросервіси підкреслюють масштабованість, адаптивність, економічну ефективність і простоту додавання нових функцій на відміну від монолітного дизайну.

Оскільки кожен сервіс функціонує як незалежний додаток, довгострокова масштабованість є основною метою мікросервісів.

Залежно від обсягу продукції та пріоритетів організації можна вибрати одну з двох стратегій.

Мікросервіси нададуть вам безсерверні мікросервіси для довгострокових рішень, якщо ви збираєтеся побудувати велику платформу, яка потребує постійного зростання.

Безсерверна архітектура становить чудовий варіант для тих, хто прагне здійснити розгортання швидко та економічно.

Список використаних джерел:

1. Сербулл М.О. Бітченко О.М Мікросервісна архітектура онлайн платформи для інвестиційного менеджменту, Харків: ХНУРЕ – 2 с.

2. Singleton, A. The Economics of Microservices. IEEE Cloud Comput. 2016 С. 16–20.

3. Li, D.; Duan, J.; Yao, Y.; Qian, S.; Zhou, J.; Xue, G.; Cao, J.; Ansari, M.D. SoDa: A Serverless-Oriented Deadline-Aware Workflow Scheduling Engine for IoT Applications in Edge Clouds. Wirel. Commun. Mob. Comput. 2022, 2022, 7862911.

4. AWS CloudWatch Walkthrough With Realtime Usecase. Dheeraj Choudhary's Blog. URL: <https://dheeraj3choudhary.com/aws-cloudwatch-walkthrough-with-realtime-usecase> (дата звернення: 05.03.2024).

ДОДАТОК Г

Слайди презентації



МІНІСТЕРСТВО
ОСВІТИ І НАУКИ
УКРАЇНИ



ХАРКІВСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

Дослідження моделі побудови мікросервісної архітектури за допомогою безсерверних обчислень

Гавриш Д.Л., ІПЗм-22-5
Науковий керівник: д.т.н проф. Смеляков К. С.



17 червня 2024

Дослідження

- **Актуальність**

Сучасні інформаційні системи потребують високої гнучкості та масштабованості. Мікросервісна архітектура, поєднана з безсерверними обчисленнями, дозволяє досягти цих цілей завдяки використанню хмарних сервісів, таких як AWS Lambda та DynamoDB.

- **Напрямок дослідження**

Дослідження присвячено розробці мікросервісної архітектури за допомогою безсерверних обчислень на платформі AWS, включаючи використання AWS Lambda, Amazon API Gateway та DynamoDB.

- **Об'єкт дослідження**

Побудова мікросервісної архітектури з використанням безсерверних обчислень на платформі AWS



Огляд літератури (аналогів)

- Основних джерела та теорії у галузі

Основні джерела включають роботи з мікросервісної архітектури та безсерверних обчислень, таких як книги "[Building Microservices](#)" та "[Serverless Architectures on AWS](#)". Теорії ґрунтуються на розподілених системах, масштабованості та управлінні ресурсами.

- Прогалини у наявних дослідженнях

Недостатньо досліджено питання оптимізації продуктивності мікросервісів у безсерверних середовищах, а також відсутні комплексні аналізи довгострокових витрат і безпеки у великих масштабованих системах.



Постановка задачі

- Проблема

Існує потреба у створенні масштабованої та гнучкої мікросервісної архітектури, яка дозволяє ефективно використовувати безсерверні обчислення для зменшення витрат і підвищення продуктивності.

- Очікувані результати

Очікується отримання архітектури, яка забезпечує високу масштабованість, гнучкість та надійність мікросервісів, що дозволить знизити витрати на інфраструктуру та оптимізувати процеси розробки і розгортання.



Методологія

- Методи дослідження

Використані методи включають теоретичний аналіз літератури, моделювання архітектури, експериментальне впровадження та тестування мікросервісів у безсерверному середовищі.

- Інструментарій та технології

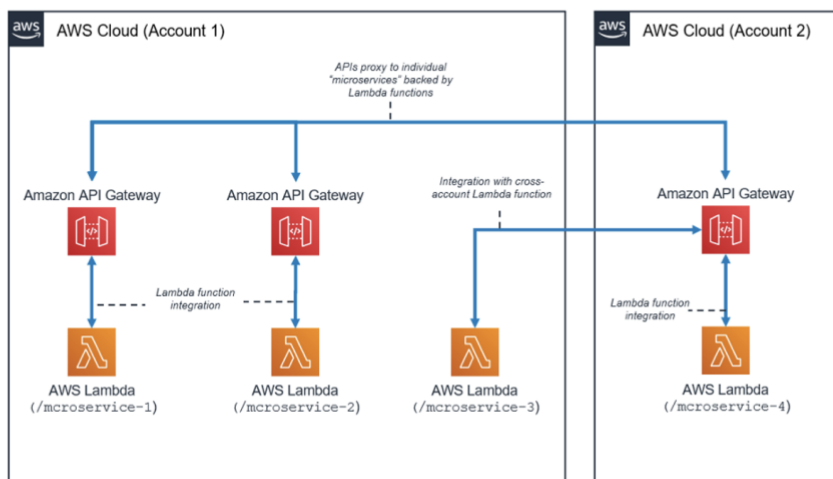
Основні інструменти: AWS Lambda, Amazon API Gateway, DynamoDB, AWS CloudWatch. Мови програмування: Python, JavaScript. Середовища розробки: AWS Management Console, Visual Studio Code.



Архітектура система для проведення експериментального дослідження

Ключові компоненти:

- **AWS Lambda**
Реалізація бізнес-логіки
- **Amazon API Gateway**
Управління API
- **DynamoDB**
Зберігання даних
- **AWS CloudWatch**
Моніторинг та логівання



Опис програмного забезпечення, що було використано у дослідженні

- Процес розробки

Процес розробки включав теоретичний аналіз, [проекткування архітектури](#), реалізацію мікросервісів на AWS Lambda, тестування та оптимізацію.

- Вибрані мови програмування та фреймворки

Python, [JavaScript](#). Фреймворки: [Serverless Framework](#), AWS SDK.



Зміст проведеного експерименту

- Методи

Теоретичний аналіз, експериментальне впровадження, тестування.

- Вхідні дані

API-запити, події AWS Lambda.

- Критерії

Продуктивність, масштабованість, витрати.

- Послідовність

Проекткування → Впровадження → Тестування → Оптимізація.

- Вимірювання

Час виконання функцій, витрати ресурсів, швидкість обробки запитів.



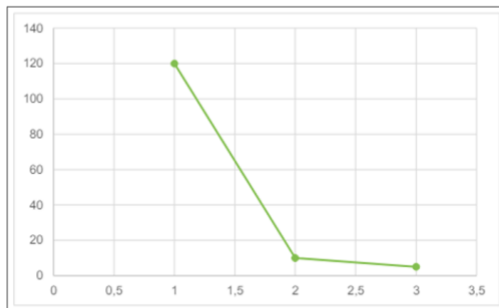
Результати експерименту

- Час затримки функцій: середнє значення 10 мс.
- Швидкість обробки даних: середнє значення 14000 операцій в мс.

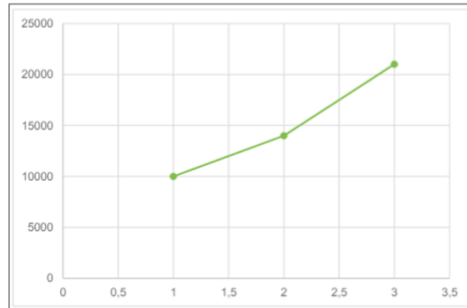
Характеристика	Позначення	Одиниця виміру	Значення параметру		
			Гірші	Середні	Кращі
Затримка роботи функцій	X1	мс	120	10	5
Швидкість обробки даних мовою програмування	X2	операції в мс	10000	14000	21000
Швидкість роботи бази даних	X3	мільйонів операцій в секунду	18	20	23
Потенційний об'єм коду	X4	рядок	4000	2500	1000



Результати експерименту



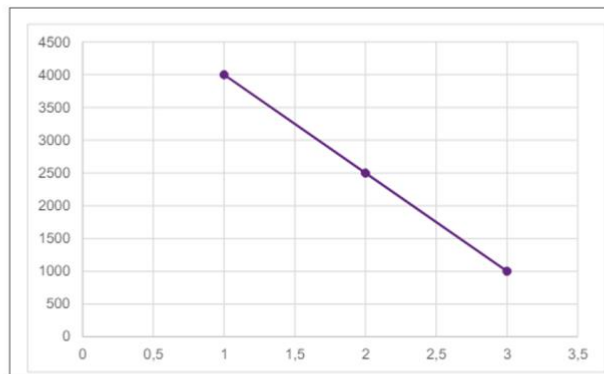
Графік X1, затримка роботи функцій



Графік X2, швидкість обробки даних мовою програмування



Результати експерименту



Графік X4, потенційний об'єм коду



Аналіз отриманих результатів

- Досягнуто високої продуктивності та масштабованості мікросервісів за допомогою AWS Lambda та інших безсерверних технологій.
- Система показала значні покращення в часі виконання та обробки даних.
- Результати підтвердили ефективність безсерверних рішень для мікросервісної архітектури.
- Дослідження демонструє перспективність використання AWS Lambda для розробки масштабованих систем.



Підсумки

- Результати показали високу продуктивність і масштабованість безсерверних мікросервісів на AWS, що підтверджує їх корисність для сучасних IT-систем. Система показала значні покращення в часі виконання та обробки даних.
- Можливість подальшого вдосконалення безпеки, оптимізації витрат і дослідження інтеграції з іншими хмарними платформами.

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на
відповідність оформлення вимогам ДСТУ 3008: 2015

Експертний висновок результатів перевірки кваліфікаційної роботи

студент
(посада)

програмної інженерії
(кафедра)

ПЗМ-22-5
(група)

Гавриш Дмитро Леонідович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні позначки	
	7.15 Додатки	

зауважень немає

Експерт

(підпис)

Олена ОЛІЙНИК

(прізвище, ініціали)

03.06.2024