

ПОБУДОВА МІКРОСЕРВІСІВ ЗА ДОПОМОГОЮ РІЗНИХ ПРОГРАМНИХ ПЛАТФОРМ

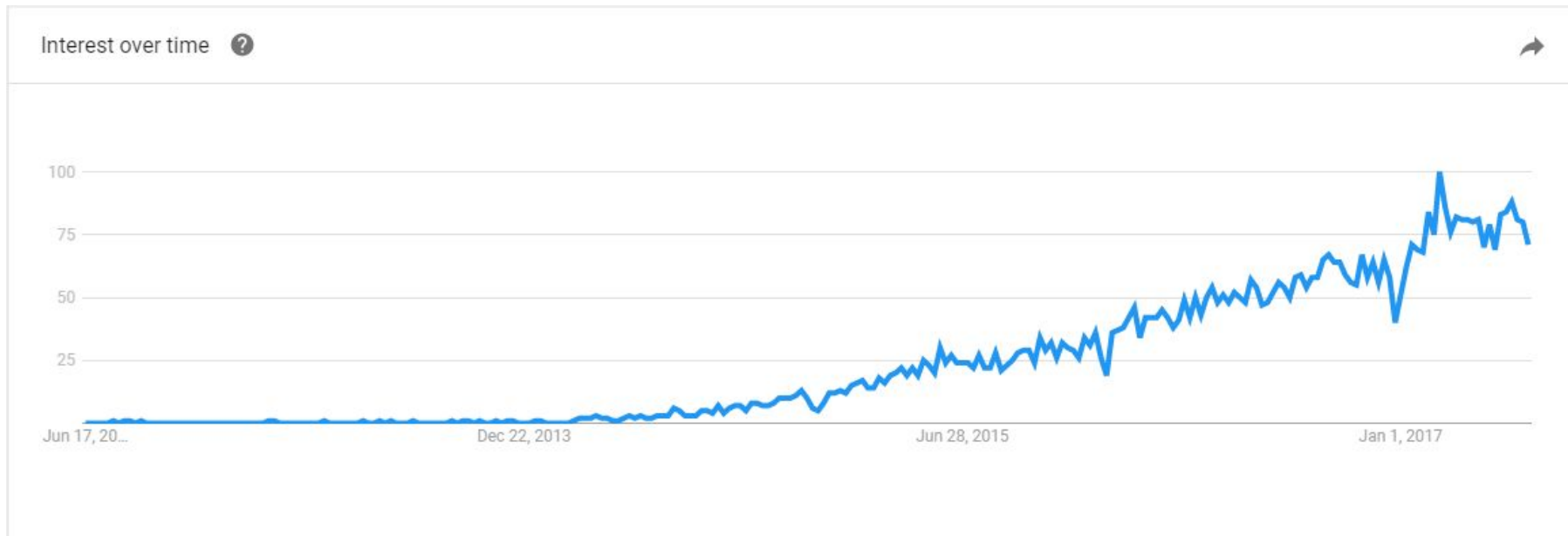
Виконав:
студент групи ДА-32
Зеленін Віктор
Науковий керівник:
Безносик Олександр Юрійович

Мета роботи

- Дослідити основні принципи побудови мікросервісних систем.
- Порівняти мікросервісну архітектуру із іншими існуючими рішеннями.
- Здійснити огляд найпопулярніших програмних платформ для побудови мікросервісів.
- Створити тестову примітивну систему на базі мікросервісів.

Актуальність теми

- Мікросервісна архітектура є сучасним підходом до створення ефективних масштабованих розподілених систем.



MSA та монолітна архітектура

MSA	Моноліт
Часткове розгортання	Простота
Відмовостійкість	Узгодженість
Відсутність стану	Міжмодульний рефакторинг
Гетерогенність	

* приклади вище вказаних архітектур наведені на плакатах #1 та #2

Взаємовідношення MSA та SOA

SOA



Microservices

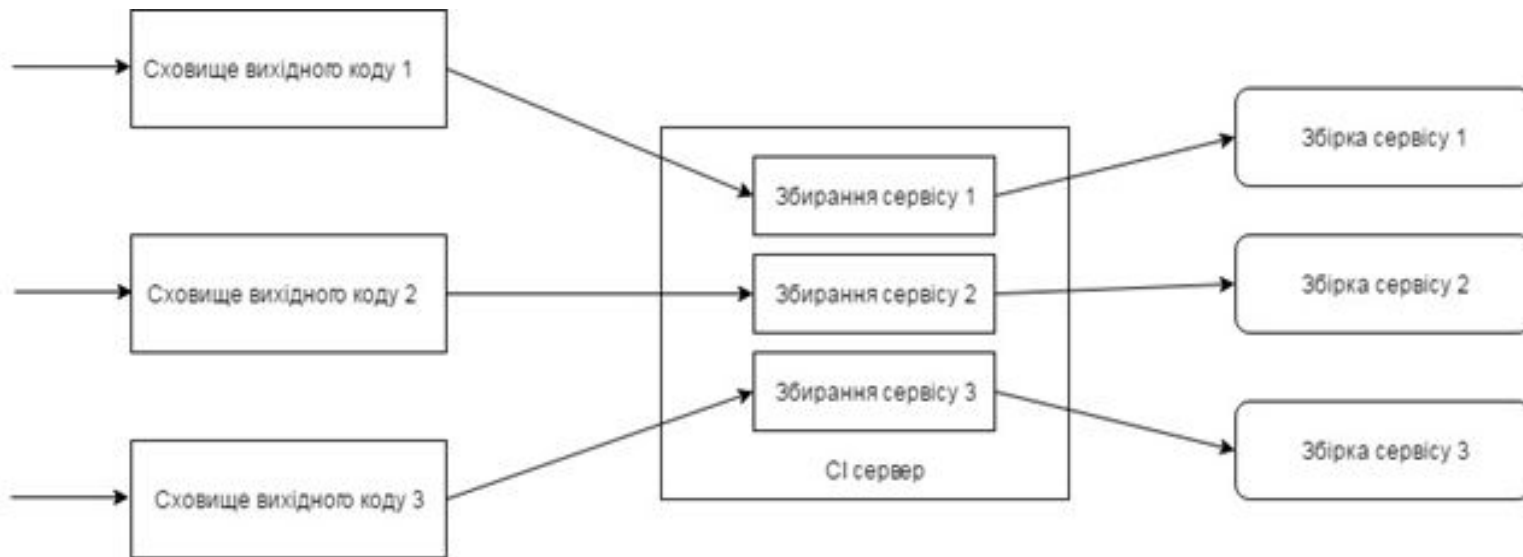


- MSA прийнято вважати наслідником SOA.
- В MSA робиться акцент на розумних приймачах та простих каналах передачі даних.

Основні компоненти мікросервісної системи

- Функціональні мікросервіси з власними сховищами даних.
- Спільна точка входу (API Gateway).
- Сервер виявлення сервісів (Service Discovery).
- Автоматичний вимикач (Circuit Breaker).

Розгортання мікросервісів



- Кожна збірка мікросервісу запускається в окремому процесі, Docker-контейнері.

Програмні платформи

В даній роботі було розглянуто платформи для наступних мов програмування:

- C++ (CppMicroServices, Pistache, ULib)
- Java (Spring framework, Spark framework, Restlet)
- Python (Flask, Tornado, Nameko)

* Порівняння платформ наведено на плакаті #3

Можливості основних платформ

Можливість	Java Spring	C++ MicroServices	Python Flask
Підтримка стандартних протоколів передачі даних	+	+	+
Якісна документація	+	-	+
Відкритий програмний код	+	+	+
Пряма інтеграція з інструментами Netflix OSS	+	-	-
Кросплатформеність	+	+	+

Тестова програмна система

Для демонстрації концепції мікросервісів було створено простий додаток, який включає в себе наступні елементи:

- Movie service
- User service
- Booking service
- API Gateway
- Service Discovery Server

* Архітектуру додатку наведено на плакаті #4

Реалізація компонентів системи

Для реалізації функціональних мікросервісів було використано:



Java Spring framework



Python Flask framework

Управління даними

В тестовій системі було використано наступні бази даних:



Результати роботи

- Було досліджено та проаналізовано мікросервісну архітектуру, основні принципи її побудови, ключові компоненти
- Розглянуто особливості MSA в порівнянні з іншими існуючими архітектурними стилями.
- Досліджено різні програмні платформи для побудови мікросервісів для найпопулярніших мов програмування.
- Створено тестовий примітивний додаток на базі мікросервісної архітектури з використанням різних мов програмування для підтвердження працездатності концепції.

Висновки

Мікросервісна архітектура не є рішенням абсолютно всіх можливих задач. Більше того, мікросервіси не рекомендовано створювати без глибокого попереднього аналізу предметної області та чіткого виділення обмежених контекстів, а також пропонується створювати мікросервіси на базі існуючого перевантаженого моноліту. Даний архітектурний стиль є молодим та перспективним у сучасному проектуванні розподілених високонавантажених систем.

ДЯКУЮ ЗА УВАГУ