

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ

Модель веб-застосунків на базі мікросервісної архітектури

Кваліфікаційна робота
Другий (магістерський) рівень

Авторка:
Осіпова Д. Ю.
студ. гр. СПм-20-2

Керівник:
Фомічов О. О.
ст. в. каф. ЕОМ

Мета і задачі роботи

Мета: аналіз можливостей мікросервісної архітектури та показати її сильні та слабкі сторони з точки зору розробки програмного забезпечення.

Задачі:

- розглянути основні протоколи, рівні мережевої взаємодії та інші механізми, які задіяні у процесі роботи веб-програми;
- розробити математичну модель розподілу користувачів у мікросервісному web-додатку
- зробити порівняльний аналіз різних типів архітектури – моноліт, SOA і мікросервісну;
- порівняльний аналіз черги повідомлень та REST API за основними критеріями оцінки ПЗ
- дослідження пропускну спроможності брокерів повідомлень Kafka і RabbitMQ
- аналіз систем оркестрації контейнерів Kubernetes, Docker Swarm та Apache Mesos

Актуальність



Типи web-додатків

- SPA, Single Page Application, односторінкові веб-програми;
- MPA, Multi Page Application, багатосторінкові веб-програми;
- PWA, Progressive Web Application, прогресивні веб-програми;
- Serverless або безсерверна архітектура;
- Архітектура мікросервісів.

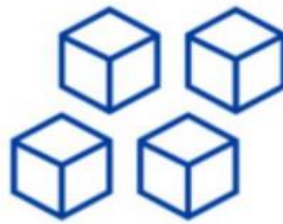
Основні типи архітектур web-додатків

Монолітна



MONOLITHIC

Сервіс-орієнтована



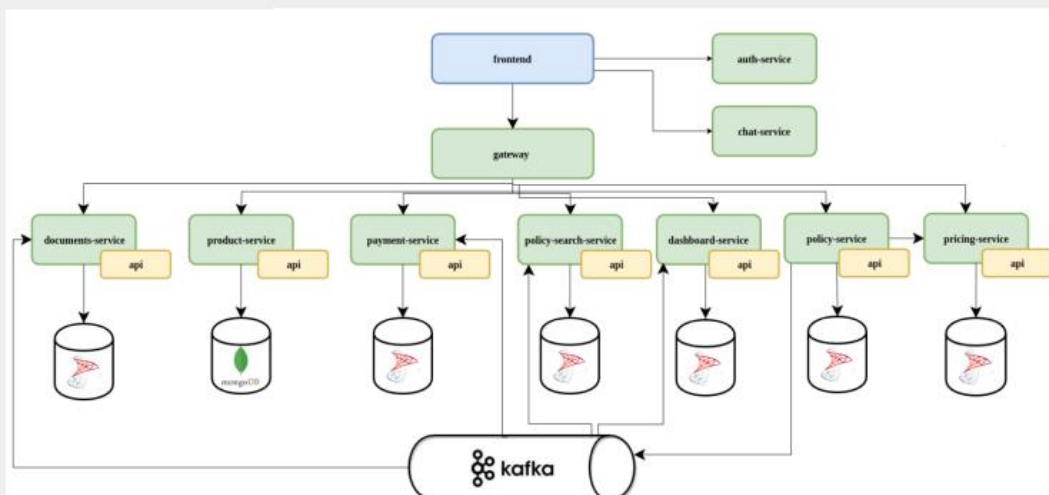
SOA

Мікросервісна



MICROSERVICES

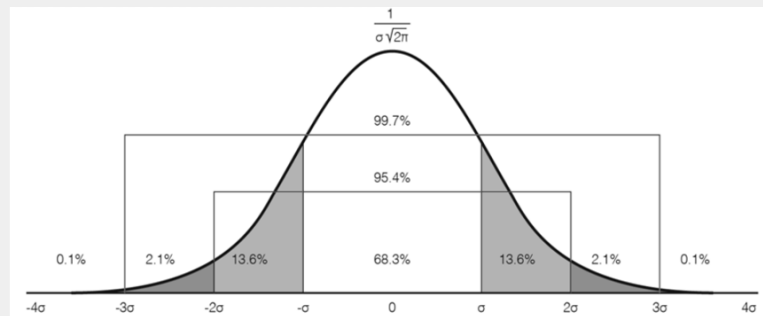
Мікросервісна архітектура



Математична модель розподілу користувачів

$$G(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}}$$

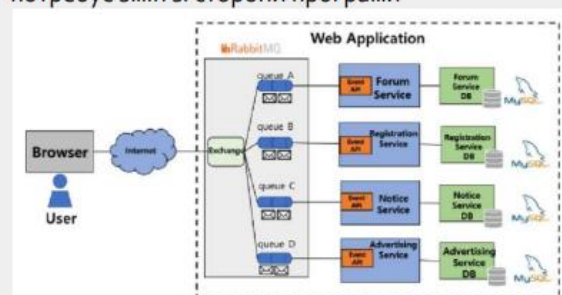
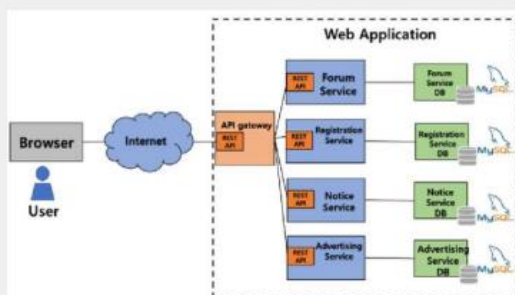
$$G(t) = \frac{4N_u}{T_{tot}\sqrt{2\pi}} e^{-\frac{(4t-3T_{tot})^2}{T_{tot}^2}}$$



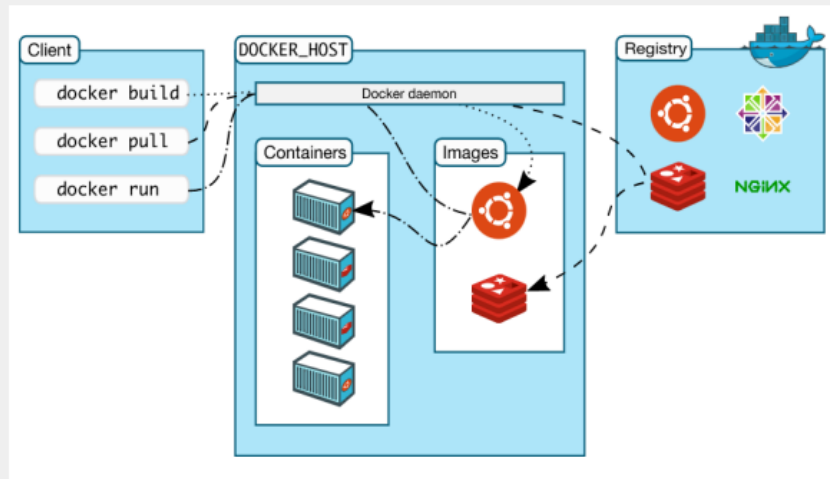
REST API & Message Queue (MQ)

- синхронний запит/відповідь – HTTP;
- загальнодоступні інтерфейси API:
- будь-яка мова програмування:
- вимагає детальної конфігурації балансувальника навантаження:
- протокол TLS.

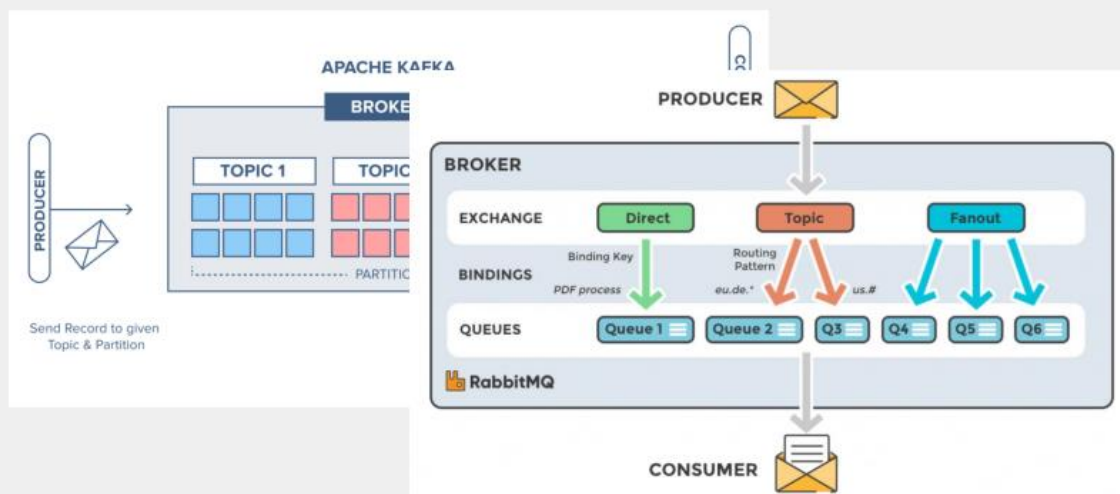
- асинхронний обмін повідомленнями;
- працює з необмеженою кількістю екземплярів
- мова програмування залежить від підтримки обраного брокера
- якщо одна система недоступна інша може продовжити обмін повідомленнями
- Використовує протокол Advance Message Security не потребує змін зі сторони програми



Розгортання з Docker

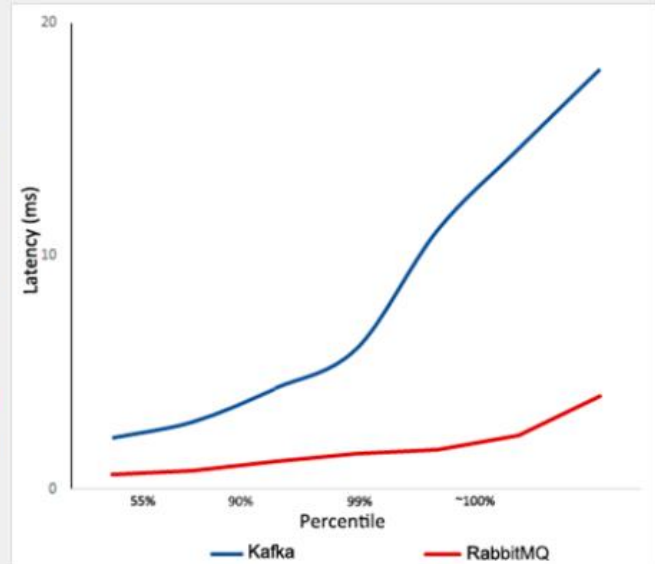


Брокери повідомлень Kafka & RabbitMQ



Експеримент

Kafka забезпечує найнижчу затримку при вищій пропускній здатності, а також забезпечує міцну довговічність і високу доступність. RabbitMQ може досягти меншої наскрізної затримки, ніж Kafka, але лише зі значно нижчою пропускнуою здатністю.



Висновки

Було проведено:

- порівняльний аналіз різних типів архітектури – моноліт, SOA та мікросервісу;
- порівняльний аналіз монолітної архітектури та мікросервісів з урахуванням основних критеріїв проектування системи;
- порівняльний аналіз черги повідомлень та REST API
- порівняльний аналіз брокерів повідомлень Kafka і RabbitMQ
- дослідження пропускної спроможності брокерів повідомлень Kafka і RabbitMQ;
- аналіз систем оркестрації контейнерів