

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.

27. Аналіз алгоритмів розподілення та управління обчислювальними ресурсами при обробці відеоданих. Денисенко А. В., Козлов О. В. (Чорноморський національний університет імені Петра Могили)	229
28. Методи розробки мобільних додатків. Дедух Т. А. (Житомирський державний університет ім. Івана Франка)	231
29. Розробка та впровадження інформаційної системи контролю руху автотранспорту. Дубина В. (Поліський національний університет)	233
30. Інформаційна система ідентифікації вибухонебезпечних предметів. Жданюк В.О., Снігур Т.С. (Одеський національний технологічний університет)	235
31. Проектування інформаційних систем і програмних комплексів. Жукова О. (Національний університет "Одеська політехніка")	237
32. Розробка інструментального засобу для автоматизованої оцінки показників якості мікросервісних застосунків. Зінов'єв Д. В., Ткачук М. В. (Харківський національний університет імені В. Н. Каразіна)	239
33. Інформаційна система управління спортивними тренуваннями на базі мобільного додатку. Іщенко Д.М., Владімірова В.Б. (Одеський національний технологічний університет)	241
34. Аналіз роботи створеного інтернет-магазину з продажу взуття. Каковкіна К.І., Корнієнко Ю.К. (Одеський національний технологічний університет)	242
35. Аналіз та перспективи розвитку меседж брокерів у мікросервісній архітектурі. Красношарпа Н.С., Селівьорстова Т.В. (Український державний університет науки і технологій)	244
36. Вимоги до засобів та методів інформаційної підтримки тренера з футболу. Кіриченко О.О. (Національний університет харчових технологій)	246
37. Розробка сервісу для написання резюме. Корнійчук М. А. (Волинський національний університет імені Лесі Українки)	247
38. Дослідження результатів впровадження інвестиційних проєктів з використанням розробленого Веб-сайту. Кюссе Є.І., Корнієнко Ю.К. (Одеський національний технологічний університет)	248
39. Дослідження інформаційних технологій діяльності волонтерських організацій. Литвиненко Г.І., Плотніков В.М. (Одеський національний технологічний університет)	250
40. Дослідження ринку праці сфери інформаційних технологій з метою виявлення пропозицій для випускників спеціальності 122 «Комп'ютерні науки». Мальцев М.С. (Одеський національний технологічний університет)	251
41. Аналіз функціоналу сервісу для сповіщення відключень електроенергії . Мартинюк В.В. (Прикарпатський національний університет імені Василя Стефаника)	253
42. Автоматизація адміністративно-управлінської діяльності у наукових та навчальних установах України. Матвейшин С.М., Петренко М.Г. (Інститут кібернетики імені В.М. Глушкова НАН)	254
43. Розвиток й перспективи ІТ технологій. Матюшков О.О., Селіванова А.В. (Одеський національний технологічний університет)	256
44. Інформаційно-управляюча система керування власним бюджетом на базі мобільного додатку. Мельников О.О., Владімірова В.Б. (Одеський національний технологічний університет)	258
45. Development of a WEB-based application for delivering the "Software testing" course. Мірошниченко Д.І., Мельник К.В., Лютенко І.В. (National Technical University "Kharkiv Polytechnic Institute")	259
46. Імплементация аналізу стандартів відкритої науки для реалізації Веб-проєктів. Мкртчян К.Р., Ольшевська О.В. (Одеський національний технологічний університет)	262

**АНАЛІЗ ТА ПЕРСПЕКТИВИ РОЗВИТКУ МЕССЕДЖ БРОКЕРІВ У
МІКРОСЕРВІСНІЙ АРХІТЕКТУРІ**
КРАСНОШАПКА Н.С. (nykytakrasnoshapka348@gmail.com), **СЕЛІВЬОРСТОВА Т.В.**
Український державний університет науки і технологій

Вступ. Мікросервіси - це архітектурний стиль, що ґрунтується на створенні незалежних служб, які можна розгортати окремо. Кожна служба організована навколо певної бізнес-області. Використовуючи ці служби, можна побудувати більшу систему, дозволяючи їм взаємодіяти через мережі. За останні роки впровадження архітектури мікросервісів сітало дедалі популярнішим. Великі компанії з високим ринковим капіталом, такі як Netflix, Spotify, Amazon і Uber, відкрито заявляють про використання цього архітектурного стилю. Оскільки мікросервіси можуть розділити моно-літний додаток на кілька незалежно розроблених та розгорнутих служб, вони надають багато переваг, зокрема: вільний зв'язок між службами, швидку доставку, збільшення масштабованості та надійності. Однак, існує багато проблем, оскільки введення мікросервісів збільшує складність системи. Збільшення складності походить з необхідності взаємодії між процесами, а також розподілу даних та управління транзакціями. Оскільки кожен мікросервіс виконується як окремий процес, необхідність у взаємодії залежить від деяких видів мережових викликів. Розподіл даних та управління транзакціями додатково ускладнюється, оскільки впровадження правильних меж транзакцій є викликом. Ця теза буде зосереджена на взаємодії між процесами (IPC) в архітектурі мікросервісів. Оскільки кожна служба працює як окремий процес, з'являється потреба взаємодії з іншими службами. Існує кілька підходів до задоволення цієї потреби. Служби можуть спілкуватися за допомогою механізмів комунікації типу запит/відповідь, які можуть бути синхронними або асинхронними.

Постановкою проблеми. При розробці системи з використанням мікросервісів стикаєшся з багатьма виборами щодо інфраструктури та інструментів. Вибір правильних парадигм та інструментів для взаємодії між сервісами може суттєво вплинути на систему. Існує багато альтернатив, кожна зі своїми характеристиками та відповідними компромісами. Ці компроміси можна оцінювати з точки зору продуктивності, легкості використання та підтримки. Через відносну молодість цього архітектурного патерну, існує мало емпіричних ресурсів, які допомагають розробникам приймати рішення щодо вибору різних механізмів IPC. Тому обрано наступне дослідницьке питання для цієї тези: "Які переваги та недоліки різних механізмів IPC в системах з використанням мікросервісів?". Метою цієї тези є допомога розробникам у виборі різних механізмів IPC. Для досягнення цієї мети була реалізована невелика система з використанням архітектурного патерну мікросервісів, та порівнюються різні механізми IPC RabbitMQ і Amazon SQS.

Робота "A Design Science Research Methodology for Information Systems Research" послужила хорошою основою для розробки невеликої системи, що складається з мікросервісів. Цю систему можна було протестувати на якісні та кількісні характеристики. Оцінюючи ці характеристики, можна було дати відповідь на початкове дослідницьке запитання: які переваги та недоліки різних механізмів IPC в системах, що використовують мікросервіси. Отже які переваги та недоліки різних механізмів міжпроцесорної взаємодії в системах, що використовують мікросервіси?". Відповіді та їх пояснення будуть надані відповідно до головних критеріїв які було порівнянно, а саме: затримка передачі, легкість використання та підтримуваність.

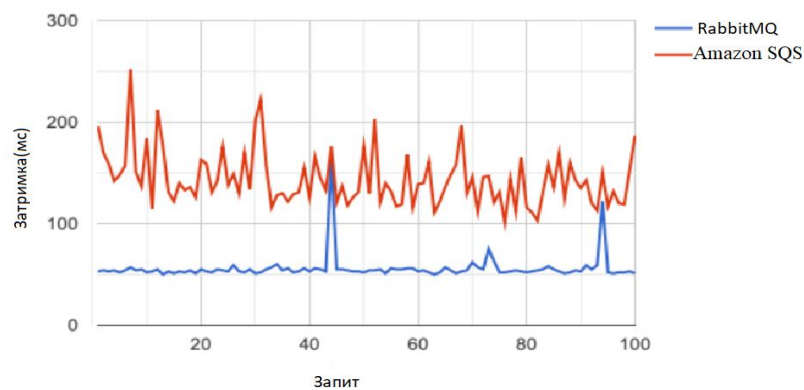
1.Затримка. Оскільки черги повідомлень базуються на деякій мережовій виклику, затримка повідомлень є аспектом, який потрібно оцінювати. Залежно від кількох факторів, таких як пропускна здатність мережі або те, чи застосунок розгорнуто локально чи віддалено, затримка повідомлень може відрізнятись. RabbitMQ та Amazon SQS показали значно відмінні результати щодо затримки. У середньому Amazon SQS мав майже втричі

вищу затримку при відправленні повідомлення через тестову систему. Оскільки Amazon SQS є віддаленим сервісом черги, де черги досягаються через Інтернет, затримка завжди буде вищою, ніж з самостійними брокерами повідомлень, такими як RabbitMQ. Причина полягає в тому, що самостійні брокери повідомлень можуть бути розгорнуті в контейнері на тому ж самому комп'ютері, що і ваші інші мікросервіси. Якщо затримка повідомлення є важливим фактором для розробки додатка, то RabbitMQ має чітку перевагу над Amazon SQS в цьому відношенні.

2.Зручність використання. Як RabbitMQ, так і Amazon SQS надають добре задокументований інтуїтивний графічний інтерфейс користувача для управління чергами та іншими адміністративними цілями. Це значно спрощує використання, оскільки черги можна додавати, видаляти та контролювати за допомогою функцій, що надаються з коробки. З цього погляду, обидва варіанти не сильно відрізняються і можуть бути вважаються сильними в цьому аспекті. Для використання черг повідомлень мікросервіси повинні бути інтегровані з брокером повідомлень через наданий API. Підтримка багатьох платформ та мов програмування є великою перевагою, оскільки це збільшує імовірність того, що більше розробників скористаються цим. З цього погляду, обидва варіанти підтримують розробку на більшості популярних платформ та мов програмування, що дає змогу вважати зручність використання сильним аспектом для обох варіантів. RabbitMQ є трохи складнішим, оскільки потрібне розуміння базового протоколу повідомлень, тоді як Amazon SQS повністю управляється та абстрагується від користувача.

3.Підтримуваність. Коли використовується брокер повідомлень, потрібно враховувати, як задовольнити угоди про рівень обслуговування з користувачами системи. У системах, що використовують самохостований брокер повідомлень, може знадобитись спеціалістична підтримка, якщо брокер не працює належним чином або допускає помилки. При використанні повністю керованої послуги можна розраховувати на хмарного провайдера для виконання угод, на яких було домовленося. Це є очевидною перевагою для Amazon SQS та подібних сервісів.

Висновок. При реалізації мікросервісної архітектури вибір правильного механізму міжпроцесного зв'язку вимагає значної уваги. Асинхронний зв'язок пропонує перевагу низько зв'язаних сервісів. Вибір повністю керованого сервісу черги повідомлень, такого як Amazon SQS, є ефективним способом реалізації асинхронного механізму міжпроцесного зв'язку з точки зору легкості використання та підтримки. Однак це супроводжується затримкою порівняно з відкритим джерелом та самохостингованим механізмом черги повідомлень, таким як RabbitMQ.



СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Peffers, Ken et al. "A Design Science Research Methodology for Information Systems Research". pp. 45–77.
2. Leung, Lawrence. "Validity, reliability, and generalizability in qualitative research". pp. 32–47.