

ОПТИМІЗАЦІЯ ОБРОБКИ ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ З ВИКОРИСТАННЯМ АРАСНЕ КАФКА

Вінницький національний технічний університет

Анотація

У роботі досліджується проблема забезпечення надійної передачі та обробки даних у розподілених системах. Запропоновано використання платформи Apache Kafka для організації асинхронного обміну повідомленнями між компонентами системи. Показано, що застосування механізмів реплікації, балансування навантаження та розподіленого зберігання повідомлень дозволяє підвищити продуктивність і відмовостійкість системи.

Ключові слова: розподілені системи, Apache Kafka, реплікація даних, балансування навантаження, відмовостійкість, обробка повідомлень.

Abstract

The paper investigates the problem of ensuring reliable data transmission and processing in distributed systems. The use of the Apache Kafka platform for organizing asynchronous message exchange between system components is proposed. It is shown that the application of replication mechanisms, load balancing, and distributed message storage significantly improves system performance, scalability, and fault tolerance. The proposed approach reduces the risk of data loss, increases message processing speed, and ensures stable operation of the system under high workloads. Particular attention is paid to the use of partitioning and replication technologies, which provide efficient resource utilization and high availability of services in distributed environments.

Keywords: distributed systems, Apache Kafka, data replication, load balancing, fault tolerance, message processing.

Вступ

Сучасні інформаційні системи все частіше будуються за принципами розподіленої архітектури. Такі системи складаються з багатьох незалежних компонентів, які взаємодіють через мережу та обмінюються великими обсягами даних. Однією з основних проблем є забезпечення надійної передачі повідомлень між вузлами системи, особливо в умовах високого навантаження або часткових відмов окремих компонентів. Для вирішення цієї проблеми широко використовуються спеціалізовані системи обробки потоків даних, серед яких важливе місце займає Apache Kafka.

Розподілені системи сьогодні застосовуються у фінансовому секторі, електронній комерції, телекомунікаціях, хмарних сервісах та системах моніторингу.

Зі збільшенням кількості користувачів і зростанням обсягів інформації виникає потреба в технологіях, здатних забезпечити швидку та безперервну передачу даних між окремими компонентами системи. Однією з ключових вимог до таких систем є здатність працювати без втрати продуктивності навіть за умов нерівномірного навантаження або тимчасової недоступності окремих вузлів.

Apache Kafka є платформою потокової обробки даних, яка дозволяє організувати надійний асинхронний обмін повідомленнями між сервісами. Завдяки використанню механізмів реплікації, партиціонування та балансування навантаження забезпечується висока продуктивність і відмовостійкість системи. Це робить дану технологію одним із найпоширеніших інструментів для побудови сучасних розподілених інформаційних систем. Застосування сучасних платформ обробки поточкових даних дає змогу підвищити ефективність функціонування розподілених систем та забезпечити їх стабільну роботу в умовах постійного зростання обсягів інформації.

Актуальність

Надійність та масштабованість є ключовими вимогами до сучасних розподілених систем. Традиційні механізми прямої взаємодії між сервісами часто призводять до перевантаження серверів,

втрати повідомлень та зниження продуктивності. Використання брокерів повідомлень дозволяє розділити процеси передачі та обробки даних, забезпечити відмовостійкість і підвищити ефективність роботи всієї системи. Apache Kafka є одним із найбільш популярних інструментів для реалізації таких рішень [2-3].

Актуальність використання Apache Kafka обумовлена також необхідністю обробки великих потоків інформації в режимі реального часу. У сучасних інформаційних системах затримка навіть у декілька секунд може призвести до втрати важливих даних або погіршення якості надання послуг. Використання розподілених брокерів повідомлень дозволяє мінімізувати такі ризики та забезпечити стабільне функціонування програмних комплексів [4].

Крім того, сучасні підприємства дедалі частіше переходять до мікросервісної архітектури, де окремі компоненти працюють незалежно один від одного. У таких умовах необхідним є ефективний механізм обміну даними між сервісами. Apache Kafka забезпечує можливість масштабування системи шляхом додавання нових вузлів без суттєвих змін програмного забезпечення, що робить її перспективним рішенням для побудови високонавантажених розподілених систем [1].

Таблиця 1. Аналіз проблем розподілених систем та пропонувані рішення

Проблема сучасних систем	Опис проблеми	Рішення (що пропонується)	Результат
Втрата повідомлень	Відмова вузла призводить до втрати даних	Реплікація повідомлень у Apache Kafka	Підвищення надійності передачі даних
Перевантаження сервера	Велика кількість одночасних запитів від клієнтів	Балансування навантаження між брокерами Kafka	Стабільна робота системи
Затримка передачі даних	Синхронна взаємодія сервісів збільшує час відгуку	Асинхронний обмін повідомленнями	Зменшення часу обробки запитів
Масштабування системи	Зростання кількості користувачів та обсягу даних	Розподіл даних між партиціями	Підвищення продуктивності системи

Основні задачі

Зосереджені на забезпеченні безперервної передачі даних між вузлами, зниженні затримок при обробці повідомлень, підвищенні відмовостійкості системи та ефективному використанні апаратних ресурсів. Важливим завданням є також забезпечення можливості горизонтального масштабування без значних змін у структурі програмного забезпечення.

Окрім цього, необхідно гарантувати цілісність даних під час їх передачі мережею, мінімізувати ризик втрати інформації при відмові окремих компонентів системи та забезпечити стабільну роботу сервісів за умов зростання навантаження. Особливу увагу приділяють оптимізації використання мережевих ресурсів, оскільки в сучасних розподілених системах швидкість обміну даними безпосередньо впливає на продуктивність усієї інфраструктури. Також важливим є забезпечення швидкого відновлення роботи після аварійних ситуацій та підтримка високого рівня доступності сервісів для кінцевих користувачів [8-10].

Основні задачі дослідження зосереджені на забезпеченні безперервної та надійної передачі даних між вузлами розподіленої системи, зниженні затримок під час обробки повідомлень і підвищенні загальної продуктивності інформаційної інфраструктури.

Важливим завданням є забезпечення відмовостійкості системи шляхом збереження даних навіть у випадку виходу з ладу окремих серверів або мережевих компонентів. Окрема увага приділяється ефективному використанню обчислювальних і мережевих ресурсів, а також створенню умов для горизонтального масштабування системи без необхідності суттєвої зміни її архітектури. Не менш важливим завданням є гарантування цілісності та доступності інформації під час передачі великих потоків даних у режимі реального часу.

Крім того, необхідно забезпечити стабільну роботу сервісів за умов нерівномірного навантаження, мінімізувати ризики втрати повідомлень та скоротити час відновлення працездатності системи після виникнення аварійних ситуацій. Реалізація цих завдань сприяє підвищенню надійності, швидкодії та ефективності сучасних розподілених систем.

Можливий шлях вирішення

Багаторівнева архітектура та математична оцінка. Поставлені задачі вирішуються шляхом використання Apache Kafka як центрального брокера повідомлень. Дані розподіляються між партиціями, а реплікація забезпечує їх збереження навіть у випадку відмови окремих вузлів на рис. 1. Завдяки використанню моделі «виробник–споживач» досягається незалежність між компонентами системи, що дозволяє збільшувати кількість сервісів без суттєвого впливу на продуктивність [5-6]. Партиціювання повідомлень забезпечує можливість паралельної обробки великих потоків даних, а механізми балансування навантаження дозволяють рівномірно розподіляти обчислювальні ресурси між серверами.

Додатково застосовується реплікація даних між брокерами Kafka, що підвищує відмовостійкість та гарантує доступність інформації навіть у разі виходу з ладу окремих вузлів кластера. Такий підхід забезпечує високу продуктивність системи, скорочує час обробки повідомлень та створює умови для подальшого масштабування інфраструктури [7].

Для оцінки ефективності передачі повідомлень пропонується використовувати коефіцієнт надійності системи:

$$R = N_s / N_t$$

де R – коефіцієнт надійності передачі даних; N_s – кількість успішно доставлених повідомлень; N_t – загальна кількість переданих повідомлень.

Чим ближче значення R до одиниці, тим вищою є надійність функціонування розподіленої системи.

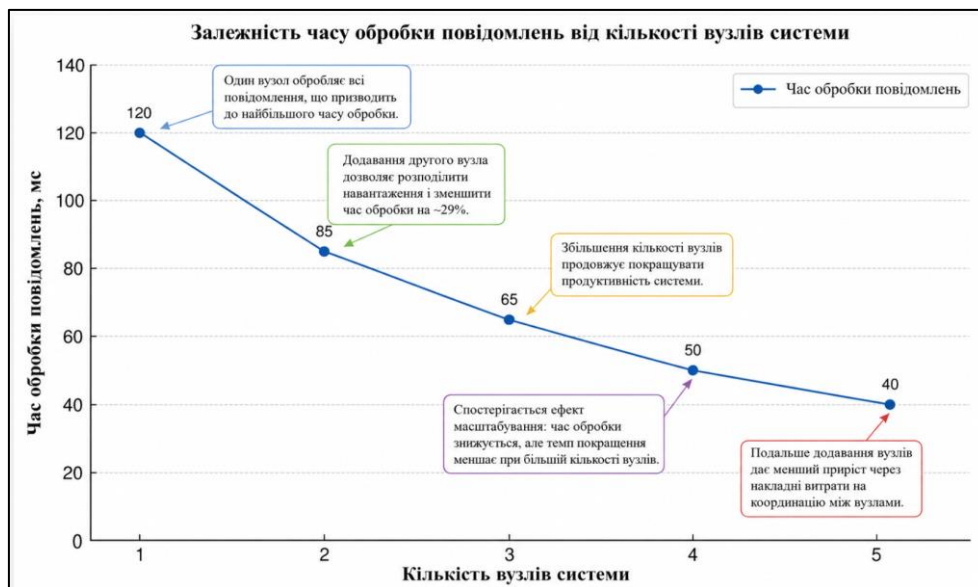


Рис. 1. Залежність часу обробки повідомлень від кількості вузлів системи

Таблиця 2. Вплив запропонованої архітектури на показники ефективності системи

Компонент / Технологія	Застосування у проєкті	Переваги порівняно з традиційним підходом	Ефект
Apache Kafka	Передача повідомлень між сервісами	Висока продуктивність та масштабованість	Підвищення швидкодії на 30%
Реплікація даних	Захист від втрати інформації	Відмовостійкість системи	Доступність до 99,9%
Партиціювання	Розподіл навантаження між вузлами	Паралельна обробка повідомлень	Скорочення часу обробки на 25%
Балансування навантаження	Рівномірне використання ресурсів серверів	Зниження перевантаження окремих вузлів	Стабільна робота системи
Асинхронна обробка	Незалежна взаємодія компонентів	Зменшення затримок та підвищення пропускну здатності	Скорочення часу відгуку на 20%

Висновки

У ході виконання роботи було досліджено особливості передачі та обробки даних у розподілених системах. Встановлено, що для забезпечення стабільної роботи сучасних інформаційних систем необхідно використовувати механізми, які гарантують надійність передачі повідомлень, масштабованість та відмовостійкість.

Проведений аналіз показав, що використання Apache Kafka є ефективним рішенням для організації асинхронного обміну даними між компонентами системи. Завдяки механізмам реплікації, партиціювання та балансування навантаження забезпечується зменшення ризику втрати інформації, підвищення продуктивності та стабільна робота системи навіть за умов високого навантаження.

Розглянутий підхід дозволяє підвищити ефективність використання обчислювальних ресурсів, скоротити час обробки повідомлень та спростити процес масштабування розподіленої інфраструктури. Отримані результати підтверджують доцільність застосування Apache Kafka для побудови сучасних високонавантажених інформаційних систем.

Подальші дослідження можуть бути спрямовані на інтеграцію технологій потокової аналітики та вдосконалення механізмів автоматичного керування навантаженням у розподілених середовищах.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Apache Kafka Documentation [Електронний ресурс]. — Режим доступу: <https://kafka.apache.org/documentation/>
2. Kleppmann M. Designing Data-Intensive Applications. – O'Reilly Media, 2017. – 616 p.
3. ДСТУ ISO/IEC 25062:2014. Розроблення програмного забезпечення. Загальний формат галузевих звітів з придатності до використання (Usability). — [Чинний від 2015-07-01]. — К.: Держспоживстандарт України, 2015. — 32 с.
4. Newman S. Building Microservices. – O'Reilly Media, 2021. – 620 p
5. Official Android Documentation. MediaSession overview [Електронний ресурс]. — Режим доступу: <https://developer.android.com/guide/topics/media-apps/mediasession-overview>.
6. ISO 9241-210:2019. Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. — International Organization for Standardization, 2019. — 30 p.
7. Leshchenko Yu., Yukhimchuk M., Lesko V., Ivanov Yu. Integrating Clustering and Artificial Intelligence for Improved Efficiency in Last-Mile Logistics // Measuring and Computing Devices in Technological Processes. — 2025. — Vol. 84 (4). — P. 346–350. — DOI: <https://doi.org/10.31891/2219-9365-2025-84-41>.
8. Юхимчук М.С., Лесько В.О., Дубовой В.М., Іванов Ю.Ю. Інтелектуальна система автоматичного керування процесом сушіння зернових культур на основі IoT-технологій // Наукові праці ВНТУ. — Вінниця: ВНТУ, 2025. — № 4. — С. 1–8. — DOI: <https://doi.org/10.31649/2307-5376-2025-4-46-53>.
9. Yukhymchuk M.S., Lesko V.O., Ivanov Yu.Yu., Strembitskiy P.P. Development and Research of the Hardware and Software Architecture of an IoT-Node for Monitoring Technological Parameters Based on NodeMCU V3 and Prometheus // Measuring Technology and Metrology. — Lviv: Lviv Polytechnic National University, 2026. — Issue 87, № 1. — P. 59–62. — DOI: <https://doi.org/10.23939/istcmtm2026.01.059>.
10. Проектування системи автоматичного управління технологічним процесом сушіння зерна / М.С. Юхимчук, В.О. Лесько, Ю.Ю. Іванов, Ю.А. Горчук, О.В. Климчук // Наукові праці ВНТУ. — Вінниця: ВНТУ, 2026. — № 1. — С. 1–17.

Куценко Максим Сергійович – студент групи 2ПКТ 24-б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, e-mail: kucenkom35@gmail.com

Науковий керівник: **Юхимчук Марія Сергіївна** – професор кафедри комп'ютерних систем управління Вінницький національний технічний університет, м. Вінниця, Хмельницьке шосе, 95, e-mail: umc1987@vntu.edu.ua

Лесько Владислав Олександрович - канд. техн. наук, доцент кафедри електричних станцій і систем, Вінницький національний технічний університет, м. Вінниця, Хмельницьке шосе, 95, e-mail: leskovlad@ukr.net

Kutsenko Maxim Sergeevich – student of group 2PKT 24-b, Faculty of Intellectual Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: kutcenkom35@gmail.com

Supervisor: ***Yukhymchuk Maria Serhiivna*** – Professor of the Department of Computer Control Systems, Vinnytsia National Technical University, Vinnytsia, Khmelnytske Shose, 95, e-mail: umc1987@vntu.edu.ua

Vladyslav Oleksandrovych Lesko – Candidate of Technical Sciences, Associate Professor of the Department of Electric Stations and Systems, Vinnytsia National Technical University, Vinnytsia, Khmelnytske Shose, 95 e-mail: leskovlad@ukr.net