

АНАЛІЗ МЕТОДІВ ПІДВИЩЕННЯ ШВИДКОДІЇ СИСТЕМ ЗБОРУ ТА ОБРОБКИ ФІНАНСОВИХ ДАНИХ У РЕАЛЬНОМУ ЧАСІ

Вінницький національний технічний університет

Анотація

У роботі досліджено методи підвищення швидкодії систем збору та обробки фінансових даних у реальному часі. Проаналізовано сучасні архітектурні підходи та технології обміну даними, зокрема асинхронну обробку запитів у Python та WebSocket-з'єднання. Показано, що застосування асинхронних механізмів і потокових протоколів значно зменшує затримки і підвищує пропускну здатність систем.

Ключові слова: фінансові дані, швидкодія, асинхронна обробка запитів, WebSocket, API.

Abstract

The paper investigates methods for improving the performance of real-time financial data collection and processing systems. It analyses modern architectural approaches and data exchange technologies, in particular asynchronous query processing in Python and WebSocket connections. It shows that the use of asynchronous mechanisms and streaming protocols significantly reduces delays and increases system throughput.

Keywords: financial data, speed, asynchronous request processing, WebSocket, API.

Вступ

У сучасних умовах розвитку фінансових послуг широкого поширення набуває процес цифрової трансформації та впровадження фінансових технологій. Фінансові і цифрові технології розширюють можливості ведення фінансового бізнесу, забезпечують доступ до послуг та сервісів банківських установ, створюють нові інструменти для електронної комерції, монетарних механізмів, оцінювання платоспроможності тощо. Крім того, постійно та стрімко зростають обсяг і швидкість передачі даних. Існуючі рішення щодо систем збору, передачі та аналітичної обробки великих обсягів фінансових даних у реальному часі потребують постійного вдосконалення.

Особливо актуальною ця проблема є у сфері криптовалют, де ціни можуть змінюватися щосекунди. Своєчасне отримання та обробка актуальних даних є критично важливими для прийняття рішень, побудови моделей прогнозування та моніторингу ринку [1]. Тому підвищення швидкодії інформаційних систем, що працюють із потоками фінансових даних, є актуальним завданням у галузі комп'ютерної інженерії.

Результати дослідження

Сучасні фінансові платформи та аналітичні сервіси ґрунтуються на розподілених програмних системах, які здійснюють збір даних із численних біржових джерел через мережеві інтерфейси. За умов високої частоти оновлення котирувань навіть незначні затримки на рівні програмної реалізації можуть призводити до втрати актуальності даних і зниження ефективності аналізу.

У цьому контексті особливого значення набуває дослідження програмних методів організації збору та обробки фінансових даних у режимі реального часу.

Аналіз існуючих методів організації збору та обробки фінансових даних у реальному часі, що використовуються у сучасних веб-орієнтованих аналітичних системах, показав, що основними факторами, які впливають на їх швидкодію, є затримка при зверненні до зовнішніх API, обмеження швидкості запитів, надлишкове дублювання даних під час агрегації, неефективне використання мережевих потоків і синхронні методи обробки даних.

Для зменшення затримок у роботі програмної системи збору та обробки фінансових даних у реальному часі доцільно використовувати асинхронну обробку запитів за допомогою Python бібліотек `asyncio` та `aiohhttp`, а також WebSocket-з'єднання, що забезпечують безперервну передачу даних без необхідності повторного відкриття каналу зв'язку [2].

Аналіз сучасних досліджень показав, що асинхронні реалізації обробки HTTP-запитів здатні обробляти суттєво більше запитів за одиницю часу та знижувати середню затримку порівняно з синхронними моделями. Згідно з результатами, наведеними в роботі [3], асинхронна реалізація за допомогою Python фреймворку FastAPI, демонструє суттєво кращі показники продуктивності порівняно із синхронною. Асинхронна версія обробляла 50,68 запитів/сек, тоді як синхронна — лише 36,89 запитів/сек, що становить на 37% більше пропускну здатності. Крім того, вона також мала на 27% меншу середню затримку відповіді при 100 одночасних підключеннях.

Використання WebSocket-з'єднань є ефективнішим для передавання даних у режимі реального часу порівняно з традиційними REST-API, які базуються на періодичному опитуванні сервера. Згідно з результатами експериментального порівняння, наведеними у роботі [4], присвяченій аналізу REST-API та WebSocket для комунікації у реальному часі, WebSocket-протокол забезпечує суттєво меншу затримку передачі повідомлень — у межах 10–50 мс, тоді як при використанні REST-підходу затримка становить 250–400 мс. Крім того, пропускну здатність WebSocket-з'єднання у тестових сценаріях була вищою до 5 разів, що пояснюється використанням постійного двонаправленого каналу зв'язку без необхідності повторного встановлення HTTP-з'єднань.

Висновки

Підвищення швидкодії систем збору та обробки фінансових даних у реальному часі значною мірою залежить від ефективності обробки запитів і мережевих протоколів.

Використання асинхронних механізмів обробки запитів дозволяє суттєво збільшити пропускну здатність і зменшити затримки відповіді, що критично для систем із високою частотою оновлення даних.

WebSocket-з'єднання забезпечує мінімальні затримки та підвищує пропускну здатність порівняно з традиційними REST-підходами, роблячи його ефективним рішенням для потокової передачі даних у режимі реального часу.

Поєднання асинхронної обробки, потокових протоколів є перспективним напрямом підвищення швидкодії фінансових аналітичних систем, здатних реагувати на зміну навантаження та мережевих умов.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Kumar S. Real-Time Data Streaming: Transforming FinTech Through Modern Data Architectures. *European Journal of Computer Science and Information Technology*. 2025. № 13(18). P. 49-64. URL: <https://eajournals.org/ejcsit/wp-content/uploads/sites/21/2025/05/Real-Time-Data-Streaming.pdf> (дата звернення: 05.01.2026).
2. Kokalko M. Using Asynchronous Programming in Python To Improve Application Performance. *The American Journal Of Engineering And Technology*. 2024. Vol. 6, Issue 12. P. 51-58. URL: <https://theamericanjournals.com/index.php/tajet/article/view/5722/5294> (дата звернення: 05.01.2026).
3. Dhruv P. FastAPI Performance Showdown: Sync vs Async — Which is Better?. *Medium*. URL: <https://thedkpatel.medium.com/fastapi-performance-showdown-sync-vs-async-which-is-better-77188d5b1e3a> (дата звернення: 05.01.2026).
4. Keshari R., Pachlasiya K. Comparing REST API vs Websockets for Chat Applications. *International Journal of Innovative Research in Science Engineering and Technology*. 2025. Vol. 14, Issue 3. P. 5009-5012. URL: https://www.ijirset.com/upload/2025/march/416_Comparing.pdf (дата звернення: 05.01.2026).

Дидяк Максим Богданович — студент групи 1KI-25м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, email: maksimdidyak234@gmail.com

Войцеховська Олена Валеріївна — кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця.

Dydiak Maksym Bogdanovych — student of group 1KI-25m, faculty of information technologies and computer engineering, Vinnytsia National Technical University, Vinnytsia, email: maksimdidyak234@gmail.com

Voitsekhovska Olena Valeriivna — PhD, Assistant Professor of the Computer Techniques Department, Vinnytsia National Technical University, Vinnytsia.