

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

"На правах рукопису"
УДК 519.86

ДО ЗАХИСТУ ДОПУЩЕНО
Завідувач кафедри

_____ Олександр РОЛІК

“ _____ ” _____ 2023 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за освітньо-науковою програмою

«Інтегровані інформаційні системи»

зі спеціальності 126 *«Інформаційні системи та технології»*

на тему:

«Інформаційна технологія обробки даних на криптобіржах»

Виконав:

студент VI курсу, групи ІА-11мн
Дуда Володимир Олександрович

Керівник:

завідувач кафедри, д.т.н., проф.,
Ролік Олександр Іванович

Рецензент:

завідувач кафедри ОТ
д.т.н, проф.,
Стіренко Сергій Григорович

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.

Студент _____

Київ – 2023 року

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-наукова програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2023 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Дуда Володимир Олександрович

1. Тема дисертації «Інформаційна технологія обробки даних на криптобіржах»

науковий керівник дисертації Ролік Олександр Іванович, д.т.н., проф.,
затверджені наказом по університету від «20» березня 2023 р. №1275-с

2. Строк подання студентом дисертації “ 12 ” травня 2023 р.

3. Об'єкт дослідження: процес для збору, аналізу, зберігання та відображення даних з криптобірж.

4. Предмет дослідження: методи аналізу, зберігання даних з криптобірж.

5. Перелік завдань, які потрібно розробити: провести огляд предметної області на можливість автоматизації процесів; перевірити доступні стратегії для торгівлі та обрати одну для подальшого дослідження; розробити алгоритм для аналізу даних з криптобірж для подальшого розрахунку керуючого параметру; розробити архітектури для збору, аналізу, зберігання та відображення даних; перевірка отримані дані на існуючому інструменті для торгівлі.

5. Перелік графічного (ілюстративного) матеріалу

1. UML діаграма компонентів
2. UML діаграма діяльності для обрахунку спредів
3. UML діаграма діяльності для збереження спредів
4. Діаграма розгортання проєкту
5. Діаграма варіантів використання проєкту
6. Головна сторінка проєкту
7. Результат використання вихідної інформації для торгівлі

6. Орієнтовний перелік публікацій

Дуда В., Ролік О. Прогнозування короткочасних даних для крипторинку // Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління». – 2023. – № 1 (42). – с. 141–152.

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання “ 31 ” січня 20 23 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Систематизація результатів огляду літератури	10.02	
2	Порівняльний аналіз існуючих інструментів	15.02	
3	Постановка та формалізація математичної моделі задачі	07.03	
4	Модифікація існуючих алгоритмів обробки даних	15.03	
5	Розробка інформаційного та програмного забезпечення	26.03	
7	Проведення експериментальних досліджень розроблених алгоритмів	29.03	
8	Оформлення документації	14.04	
9	Подання роботи на попередній захист	20.04	
10	Подання роботи на основний захист	17.05	

Студент _____

Володимир ДУДА

Науковий керівник _____

Олександр РОЛІК

РЕФЕРАТ

Магістерська дисертація: 111 с., 54 рис., 6 табл., 32 джерел, 7 додатків.

Актуальність. За останні роки ринок криптовалют значно збільшив вплив на світову економіку. Як результат – можливість використання їх для торгівлі та платежів. Криптовалюти не залежать від фінансових установ, таких як банки, і це дозволяє людям володіти та контролювати свої фінанси безпосередньо самим.

Навіть в скрутний час для України значну частину донатів отримано саме в криптовалюті.

Для ефективного використання криптоактивів потрібно аналізувати зміни цін з різних джерел. Слідкуючи за трендом ціни, можна купляти дешевше та продавати дорожче і, в результаті чого, отримувати збільшення кількості криптовалютних монет.

Мета дослідження – підвищення швидкості обробки даних з криптобірж з покращенням пропускної можливості та підвищенням надійності зберігання.

Для досягнення мети необхідно виконати наступні завдання:

- провести огляд предметної області на можливість автоматизації процесів;
- перевірити доступні стратегії для торгівлі та обрати одну для подальшого дослідження;
- розробити алгоритм для аналізу даних з криптобірж для подальшого розрахунку керуючого параметру;
- розробити архітектури для збору, аналізу, зберігання та відображення даних;
- перевірка отримані дані на існуючому інструменті для торгівлі.

Об'єкт дослідження – процес для збору, аналізу, зберігання та відображення даних з криптобірж.

Предмет дослідження – методи аналізу, зберігання даних з криптобірж.

Наукова новизна одержаних результатів полягає в вдосконаленні архітектури та алгоритмів обробки, зберігання даних з криптобірж для ефективної та безпечної торгівлі на криптобіржах.

Публікації. Дуда В., Ролік О. Прогнозування короткочасних даних для крипторинку // Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління». – 2023. – № 1 (42). – с. 141–152.

КРИПТОВАЛЮТИ, КРИПТОБІРЖІ, СТРАТЕГІЇ ТОРГІВЛІ,
ПРОГНОЗУВАННЯ ТРЕНДУ, ЕКСПОНЕНЦІЙНЕ ЗГЛАДЖУВАННЯ,
РОЗРАХУНОК СПРЕДУ, ВИКОРИСТАННЯ СПРЕДУ ДЛЯ ТОРГІВЛІ

ABSTRACT

Master's dissertation: 111 pp., 54 figs., 6 tables, 32 sources.

Topicality. In recent years, the cryptocurrency market has significantly increased its influence on the global economy. As a result, it is possible to use them for trade and payments. Cryptocurrencies are independent of financial institutions such as banks, allowing people to own and control their finances directly.

Even in difficult times for Ukraine, a significant part of donations was received in cryptocurrency.

To effectively use crypto assets, you need to analyze price changes from various sources. By following the price trend, you can buy cheaper and sell more expensive and, as a result, receive an increase in the number of cryptocurrency coins.

The purpose of the study – increasing the speed of data processing from crypto exchanges with improved throughput and increased storage reliability.

To achieve the goal, it is necessary to complete the following **tasks**:

- conduct a review of the subject area for the possibility of automating processes;
- check the available strategies for trading and choose one for further research;
- develop an algorithm for analyzing data from crypto exchanges for further calculation of the control parameter;
- develop architectures for data collection, analysis, storage and display;
- verification of received data on an existing trading instrument.

The object of research – a process for collecting, analyzing, storing and displaying data from crypto exchanges.

The subject of research – methods of analysis, storage of data from crypto exchanges.

Publications. Duda V., Rolik O. Forecasting of short-term data for the crypto market // Interdepartmental scientific and technical collection "Adaptive systems of automatic control". – 2023. – No. 1 (42). - pp. 141–152.

CRYPTOCURRENCIES, CRYPTO EXCHANGE, TRADING STRATEGIES,
TREND FORECASTING, EXPONENTIAL SMOOTHING, SPREAD
CALCULATION, USING THE SPREAD FOR TRADING

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ.....	11
ВСТУП	12
1 ОГЛЯД СУЧАСНИХ ДОСЛІДЖЕНЬ ТА РІШЕНЬ В ОБЛАСТІ ТОРГІВЛІ КРИПТОВАЛЮТАМИ.....	14
1.1 Постановка задачі.....	14
1.2 Огляд криптовалют та криптомереж	15
1.3 Огляд сучасних криптобірж.....	16
1.4 Дослідження недоліків та ризиків крипторинку.....	17
1.5 Огляд інструментів для торгівлі на криптобіржі.....	18
1.6 Огляд торгівлі ф'ючерсними контрактами та опціонами.....	19
1.7 Огляд ф'ючерсної позиції	21
1.8 Огляд операцій для торгівлі з біржею	22
1.9 Огляд ордербуку та трейдів	25
1.10 Огляд популярних стратегій торгівлі.....	27
1.11 Торгівля за спредом	29
1.12 Алгоритм торгівлі за спредом.....	31
1.13 Математичний зміст спреду.....	32
1.14 Огляд сучасних інструментів для торгівлі	33
1.14.1 TradingView.....	34
1.14.2 Delta	35
1.14.3 Coinigy	36
1.14.4 3Commas	37
1.14.5 TradeSanta.....	38

	8
1.14.6 Hummingbot.....	40
1.14.7 Kryll.io.....	40
1.15 Огляд досліджуваної криптобіржі.....	41
1.16 Огляд інструменту для перевірки отриманих значень спреду на криптобіржі	43
Висновки до розділу	43
2 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ДЛЯ ЗБОРУ, АНАЛІЗУ, ЗБЕРІГАННЯ ТА ВІДОБРАЖЕННЯ ДАНИХ З КРИПТОБІРЖ	44
2.1 Способи збору даних з криптобірж.....	44
2.2 Огляд видів даних з криптобірж.....	45
2.2.1 Формат даних тикерів	46
2.2.2 Формат даних ордербуку.....	47
2.2.3 Формат даних трейдів	48
2.3 Способи покращення обробки даних з криптобірж	49
2.4 Паралельна потокова обробка	50
2.5 Агрегація та перетворення даних	52
2.6 Використання брокерів повідомлень	53
2.7 Забезпечення надійності збору даних.....	54
2.8 Особливості роботи брокерів повідомлень	56
2.9 Системи асинхронного збереження даних	58
2.10 Опис бізнес – процесів.....	60
2.10.1 Опис процесу діяльності.....	60
2.10.2 Актори і функції	62
2.10.3 Структура бізнес-процесів	63
2.11 Постановка задачі дослідження.....	63
2.12 Розробка компонентів системи.....	64

	9
2.12.1 Компонент збору даних та обчислення спреду.....	64
2.12.2 Компонент запису даних	66
2.13 Рішення з інформаційного забезпечення.....	67
Висновки до розділу	68
3 ПІДХОДИ ДО ЗГЛАДЖУВАННЯ СПРЕДУ ЦІНИ.....	70
3.1 Постановка задачі згладжування спреду цін.....	70
3.2 Аналіз методів згладжування.....	70
3.2.1 Simple Exponential Smoothing.....	70
3.2.2 Holt's Linear Method	71
3.2.3 Additive Damped Trend Method	72
3.3 Результати досліджень ефективності методів.....	73
Висновки до розділу	76
4 РОЗРОБКА ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ ...	77
4.1 Вимоги до програмного продукту.....	77
4.2 Засоби розробки	77
4.2.1 VS Code.....	77
4.2.2 Node.JS.....	78
4.2.3 WebSockets	79
4.2.4 RabbitMQ	80
4.2.4 ClickHouse	81
4.2.5 Grafana	83
4.3 Архітектура програмного забезпечення	84
4.4 Інструкція користувача.....	85
4.4.1 Підготовка середовища для сервісу зберігання спредів. Встановлення g++ компілятора.....	85
4.4.2 Встановлення rabbitmq-с.....	85

	10
4.4.2 Встановлення clickhouse-cpp.....	86
4.4.3 Компіляція сервісу збереження даних	86
4.4.4 Підготовка середовища для сервісу завантаження цін та обчислення спредів. Встановлення Node.js	88
4.4.5 Налаштування та запуск сервісу завантаження цін та обчислення спредів.....	88
4.4.5 Налаштування Grafana для відображення спредів.....	90
Висновки по розділу	95
ВИСНОВКИ.....	97
ПЕРЕЛІК ПОСИЛАНЬ	100
ДОДАТОК А Графічний матеріал.....	104
Плакат 1 UML діаграма компоненті.....	105
Плакат 2 UML діаграма діяльності для обрахунку спредів	106
Плакат 3 UML діаграма діяльності для збереження спредів	107
Плакат 4 Діаграма розгортання проєкту	108
Плакат 5 Діаграма варіантів використання проєкту.....	109
Плакат 6 Головна сторінка проєкту.....	110
Плакат 7 Результат використання вихідної інформації для торгівлі .	111

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

Трейдер	Особа або компанія, які займаються торгівлею на фінансових ринках з метою отримання прибутку.
Криптовалюти	Цифрові або віртуальні валюти, які використовуються для здійснення електронних платежів та зберігання активів.
Криптомережі	Децентралізовані системи, які використовують криптографію для надсилання транзакцій.
Ф'ючерси	Стандартизовані фінансові договори, які зобов'язують сторони укласти угоду на придбання або продаж певного активу за фіксовану ціну в майбутньому на певну дату.
Ліквідність	Здатності активу, ринку або компанії бути перетвореним на готівку швидко і без втрати значної вартості. Це міра того, наскільки легко актив можна продати або обміняти на інші активи без зміни його ціни.
Опціони	Фінансові похідні інструменти, що надають право, але не зобов'язують покупця опціону купити або продати певний протягом певного періоду часу або на певну дату в майбутньому. При цьому продавець опціону зобов'язується продати або купити актив, якщо покупець опціону вирішить його використати.

ВСТУП

За останні роки ринок криптовалют значно зріс, і в результаті чого стало можливе використання їх для торгівлі та платежів, що дозволяє будь-кому стати «гравцем» на глобальному економічному ринку. Криптовалюту можна купувати та продавати в будь-який час та в будь-якій точці світу, що робить торгівлю криптовалютами швидкою та доступною кожному. Криптовалюти не залежать від фінансових установ, таких як банки, і це дозволяє людям володіти та контролювати свої фінанси безпосередньо.

Обробка та зберігання даних з криптобірж дозволяє проводити аналіз тенденцій та попиту на різні криптовалюти, що може допомогти в розумінні ринку та прийнятті більш обґрунтованих торговельних рішень. Щоб забезпечити експертів якісними даними для роботи з торговельними стратегіями буде розроблена система, яка буде отримувати данні з криптобірж, аналізувати їх і потім зберігати їх до сховища. В результаті чого експерти через зручний інтерфейс будуть мати змогу налаштовувати торговельну систему під тенденції ринку. Як результат, буде отримано безперебійну та надійну систему, з можливістю до розширення на обробку значної кількості комбінацій торговельних пар.

У першому розділі проведений огляд сучасних досліджень та рішень в області торгівлі криптовалютами. Розглядаються постановка задачі, огляд криптовалют і криптомереж, сучасні криптобіржі, недоліки та ризики крипторинку, інструменти для торгівлі, ф'ючерсні контракти, операції на біржі, ордербук та трейди, популярні стратегії торгівлі, торгівля за спредом, алгоритм торгівлі за спредом, математичний зміст спреду, сучасні інструменти для торгівлі та огляд досліджуваної криптобіржі

Другий присвячений проектним рішенням з розробки системи для збору, аналізу, зберігання та відображення даних з криптобірж. Розглядаються способи збору даних, види даних з криптобірж, покращення обробки даних, паралельна потокова обробка, агрегація та перетворення даних, використання брокерів

повідомлень, надійність збору даних, брокери повідомлень, опис бізнес-процесів, постановка задачі, рішення з інформаційного.

У третьому розділі розглядаються підходи до згладжування спреду ціни. Поставлено задачу, розглянуті методи згладжування, такі як просте експоненціальне згладжування, і представлені результати досліджень ефективності цих методів.

У четвертому розділі розроблено програмне та технічне забезпечення для реалізації системи збору, аналізу, зберігання та відображення даних з криптобірж.

1 ОГЛЯД СУЧАСНИХ ДОСЛІДЖЕНЬ ТА РІШЕНЬ В ОБЛАСТІ ТОРГІВЛІ КРИПТОВАЛЮТАМИ

1.1 Постановка задачі

Обробка даних на криптобіржах включає в себе широкий спектр процесів, пов'язаних з збором, обробкою та аналізом даних, які виникають під час роботи з криптобіржами [3]. Ці процеси є критично важливими для успішного функціонування торговельних систем пов'язаних з криптовалюотними ринками та забезпечення ефективності торгівлі.

З задачі обробки даних можна виокремити декілька етапів:

– збір даних – криптобіржі надають різноманітні дані про торгові пари, такі як ціни, обсяги торгів, історичні дані тощо; інформація може бути отримана через публічні API, які надають доступ до даних криптобірж;

– обробка отриманих даних – після збору даних вони потребують обробки для подальшого використання; цей етап включає перевірку та очищення даних, а також може включати конвертацію до зручного формату, нормалізацію, агрегацію та інші операції, які допомагають аналізувати ринок, враховуючи його особливості;

– статистичний аналіз даних – цей етап може включати використання різноманітних статистичних методів, математичних моделей та алгоритмів машинного навчання для виявлення трендів, шаблонів та інших закономірностей, які можуть бути використані для прийняття рішень в торгівлі.

Після усіх етапів обробки дані можуть бути використані для прийняття обґрунтованих торгових рішень. Існує безліч різних торгових підходів, і кожен з них має свої переваги та недоліки. Важливо обрати підхід, який відповідає наявному капіталу, варіанту торгівлі та допустимим ризикам. Не існує універсальної стратегії, яка підходить для всіх і тому, враховуючи різні особливості ринку, можна використовувати різні стратегії.

1.2 Огляд криптовалют та криптомереж

Криптовалюти [4] – це цифрові або віртуальні валюти, які використовуються для здійснення електронних платежів та зберігання активів. За основу була взята технологія блокчейн, яка забезпечує безпечну та децентралізовану систему для обміну та зберігання.

Криптовалюти (рисунок 1.1) існують тільки в електронному вигляді та не контролюються центральними банками або законодавчими органами. Їх діяльність контролюється мережами користувачів, які взаємодіють між собою.

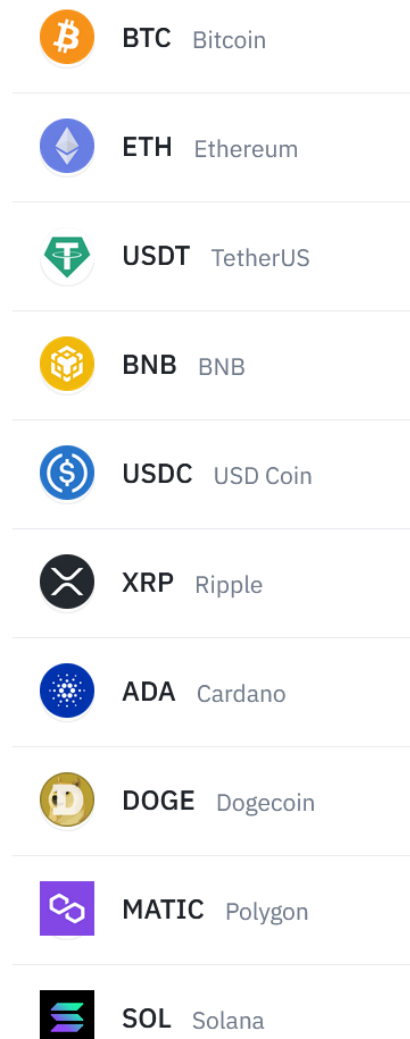


Рисунок 1.1 – Популярні криптовалюти за рейтингом Binance

Криптовалюти мають кілька переваг [5] над звичайними валютами, цінними паперами, цінними металами, газом та бензином:

– децентралізованість – операції з криптовалютами в криптомережах проводяться безпосередньо між користувачами, без брокерів, банків або інших посередників.

– анонімність – користувачі можуть зберігати дані про себе в секреті, і в транзакції будуть видні лише ідентифікатори гаманців відправника та одержувача;

– доступність – криптовалюти доступні для всіх, незалежно від місця проживання та соціального стану;

– відкритість – код криптомереж зазвичай у відкритому доступі, тобто будь хто може підняти чи власну ноду, чи навіть власну мережу.

1.3 Огляд сучасних криптобірж

Більшість криптобірж (рисунок 1.2) базуються на технології блокчейн та пропонують торгівлю криптовалютами [6], які є цифровими активами. Це означає, що криптобіржі працюють в децентралізованому середовищі, де операції проводяться без прямого втручання посередників, таких як банки чи фінансові установи.











# ▲	Name	Trading volume(24h)	Maker Fees	Taker Fees	Open Interests	No. Markets	Launched
1	 Binance	\$41,736,704,345 ▲ 7.27%	0.02%	0.04%	\$59,755,419,156	253	Jul 2017
2	 Bybit	\$10,130,333,507 ▲ 6.62%	0.01%	0.06%	\$4,344,649,281	219	Mar 2018
3	 OKX	\$14,471,657,373 ▲ 18.15%	0.02%	0.05%	\$3,315,422,059	178	Jan 2017
4	 KuCoin	\$2,147,524,776 ▲ 15.71%	0.02%	0.06%	\$1,638,746,583	150	Aug 2017
5	 Deribit	\$667,782,464 ▲ 25.29%	0%	0.05%	\$1,331,443,607	23	Jun 2016
6	 Bitget	\$7,417,122,136 ▲ 6.66%	0.02%	0.06%	\$2,785,722,366	158	Apr 2018
7	 Bitfinex	\$153,022,576 ▼ 0.98%	0.02%	0.065%	\$96,188	31	Oct 2012
8	 Kraken	\$103,793,738 ▲ 25.61%	0.02%	0.05%	\$51,797,010	104	Jul 2011
9	 BitMEX	\$740,810,588 ▲ 51.82%	0.01%	0.075%	\$2,003,148,096	28	Apr 2014
10	 Huobi	\$1,878,194,230 ▼ 2.04%	0.02%	0.04%	\$210,809,066	93	Sep 2013

Рисунок 1.2 – Популярні криптобіржі за рейтингом CoinMarketCap

Отже, такі відмінності від фондових:

- швидкість – без посередників та все автоматизовано працює 24/7; вони не мають традиційного графіка роботи, який пов'язаний з роботою в будні дні та певний годинний діапазон;
- відкритість та глобальність – доступні для всіх користувачів з різних країн світу;
- розмаїтість – можливість торгувати різними криптовалютами, що дозволяє інвесторам диверсифікувати свій портфель;
- технічні проблеми – мають свої особливості з точки зору технічної інфраструктури та безпеки; порівняно з традиційними біржами, вони частіше стикаються з проблемами технічних збоїв, кібератаками та іншим;
- менша регуляція – мають меншу регуляторну рамку порівняно з традиційними фінансовими ринками; це може призводити до більшої нестабільності та непередбачуваності на ринку.

1.4 Дослідження недоліків та ризиків крипторинку

Можливі проблеми крипторинку [7]:

- нестабільність – криптовалюти можуть бути дуже нестабільними і сильно коливатися в ціні;
- кібербезпека – криптопроекти можуть бути підвергнуті кібератакам та крадіжкам, що може призвести до втрати коштів інвесторів та користувачів;
- використання для незаконних цілей;
- проблема нераціонального використання енергоресурсів – добування криптовалют вимагає великої кількості енергії, що призводить до збільшення споживання енергії і, як наслідок – забруднення довкілля;
- регулювання – багато держав не мають чітких правових інструментів для їх регулювання.

Однак, криптовалюти – відносно новий інструмент, і багато з цих питань все ще потребують детального дослідження та вивчення [8].

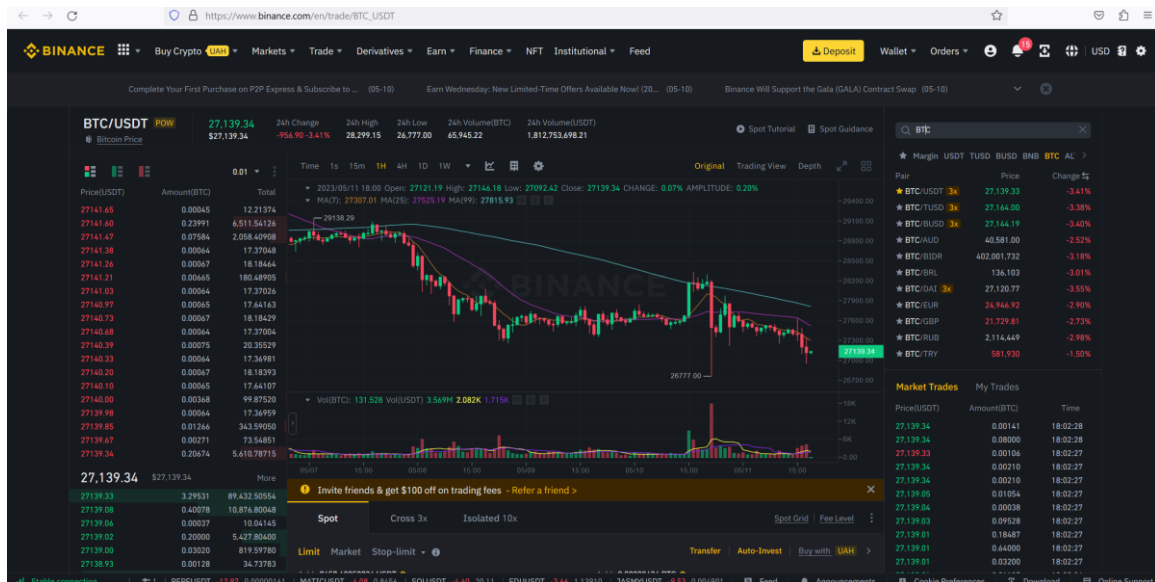
Для зменшення ризиків при торгівлі криптовалютами та їх похідними важливо дуже ретельно вивчати ринок та зберігати свої кошти на надійних і безпечних гаманцях.

Торгівля криптовалютами та ф'ючерсними контрактами може бути вигідною, але потребує великої обачності та ретельного вивчення ринку та ризиків.

1.5 Огляд інструментів для торгівлі на криптобіржі

На криптобіржах доступний ряд торговельних інструментів, які дозволяють трейдерам купувати, продавати та торгувати криптовалютами та іншими активами [9]. Розглянемо декілька популярних торговельних інструментів на криптобіржі:

– спотова торгівля – криптобіржі дозволяють трейдерам купувати та продавати різні криптовалюти, такі як Bitcoin (BTC), Ethereum (ETH), та багато інших; торгівля криптовалютами може відбуватися як парою криптовалюта/криптовалюта (BTC/USDT) (рисунк 1.3), так і парою криптовалюта/фіатна валюта (BTC/USD);



Рисунк 1.3 – Спотова торгівля на криптобіржі Binance

– ф'ючерсна торгівля та опціони – ці похідні інструменти дають трейдерам можливість укладати угоди на певні криптовалюти за певну ціну в майбутньому;

вони можуть бути використані для спекуляції на рухах цін, захисту від ризиків або реалізації інших стратегій;

– маржинальна торгівля – деякі криптобіржі надають можливість торгувати на маржі, дозволяючи трейдерам використовувати позичені кошти для збільшення своїх торгівельних позицій; це може підвищити потенційні прибутки, але також належним чином збільшує ризик втрат.

1.6 Огляд торгівлі ф'ючерсними контрактами та опціонами

Ф'ючерси на криптобіржах [11] – це контракти, які дозволяють покупцеві купувати або продавати криптовалюту за фіксовану ціну в майбутньому. Ці контракти дозволяють інвесторам та трейдерам захистити свої інвестиції від коливань цін. Крім того, ф'ючерсні контракти (рисунк 1.4) дозволяють інвесторам та трейдерам отримувати прибуток на коливаннях цін криптовалют навіть тоді, коли ринок падає.

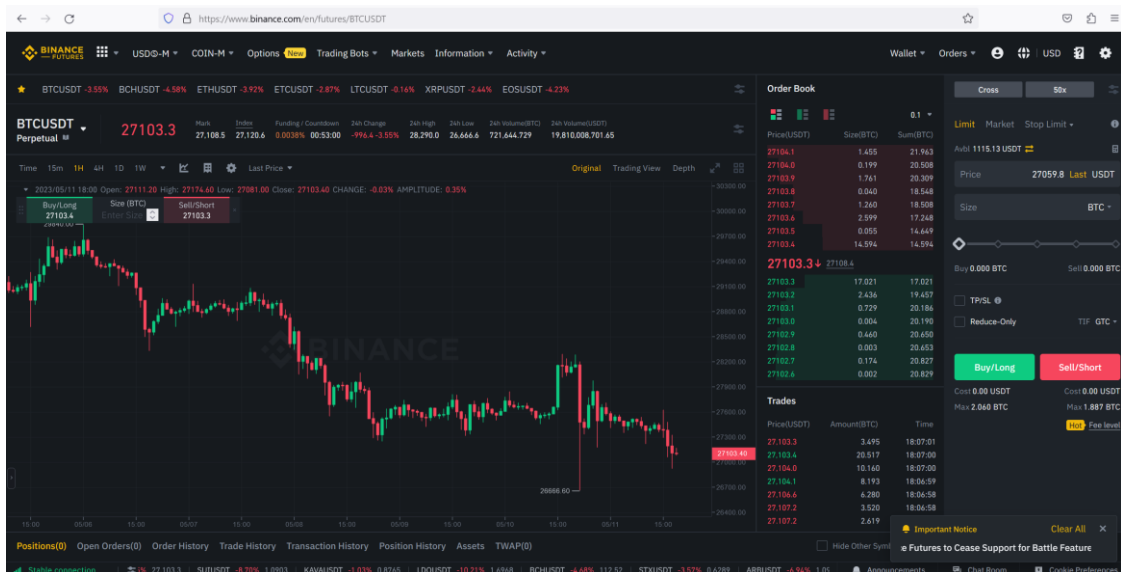


Рисунок 1.4 – Ф'ючерсна торгівля на криптобіржі Binance

Завдяки значній ліквідності та попиту на ф'ючерсні контракти криптоінвестори мають змогу входити та виходити з позицій швидко та з легкістю, зокрема завдяки великому обсягу угод на біржі.

Іншою перевагою ф'ючерсів на криптобіржах є можливість застосування різноманітних торгівельних стратегій, таких як хеджування, арбітраж та спекуляції на курсі криптовалют. Хеджування [10] (одночасна купівля та продаж

у різних місцях) може допомогти інвесторам захистити свої інвестиції від обезцінювання, так як ціна завдяки цій стратегії фіксується.

Крім того, ф'ючерси на криптобіржах можуть бути вигідні для тих, хто має не так багато капіталу, тому що вони можуть дозволити здійснювати торгівлю з використанням великих позик та кредитних рахунків. Таким чином, ф'ючерси можуть забезпечувати доступ до більшого обсягу торгівлі та бути вигідними для професійних трейдерів.

Маржа (кредитне плече) в торгівлі криптовалютами використовується для збільшення потенційного прибутку шляхом використання позичених коштів. Це дозволяє трейдерам відкривати позиції, які перевищують їх власний капітал. Однак, використання маржі також збільшує ризик втрат. Трейдери можуть позичати кошти від біржі або від інших трейдерів на платформі для збільшення свого капіталу. Криптобіржі встановлюють маржеві вимоги (рисунок 1.5), які вказують мінімальний обов'язковий розмір маржі, який треба мати на рахунку для відкриття та утримання позиції. Це забезпечує, що трейдер має достатньо капіталу для покриття можливих збитків.

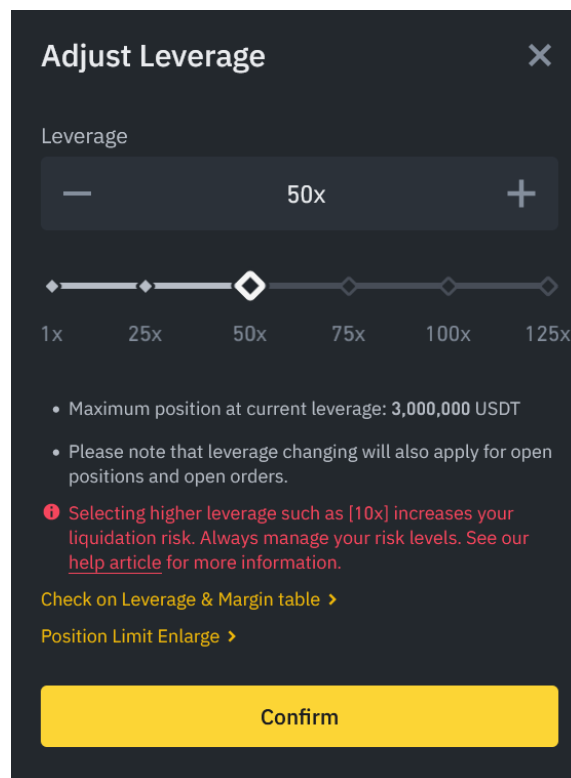


Рисунок 1.5 – Керування кредитним плечем на криптобіржі Binance для пари BTCUSDT

Торгівля ф'ючерсами на криптобіржах також має свої ризики, які пов'язані з волатильністю криптовалют та можливістю великих збитків з використанням позикових рахунків. Торгівля ф'ючерсами вимагає від трейдерів вміння прогнозувати рухи ринку та розуміння технічних та фундаментальних аспектів торгівлі.

Криптовалютні опціони – це похідні фінансові інструменти, які надають власнику право, але не обов'язок, купувати або продавати криптовалюту за певну ціну (відому як ціна виконання або страйк) до певної дати (відомої як термін дії опціону).

Опціони (рисунок 1.6) дозволяють трейдерам отримувати вигоду від коливань цін криптовалют без необхідності фактичного володіння активом. Вони можуть бути використані для різних цілей, таких як захист від ризику, спекуляція на ринку та стратегічне управління портфелем.

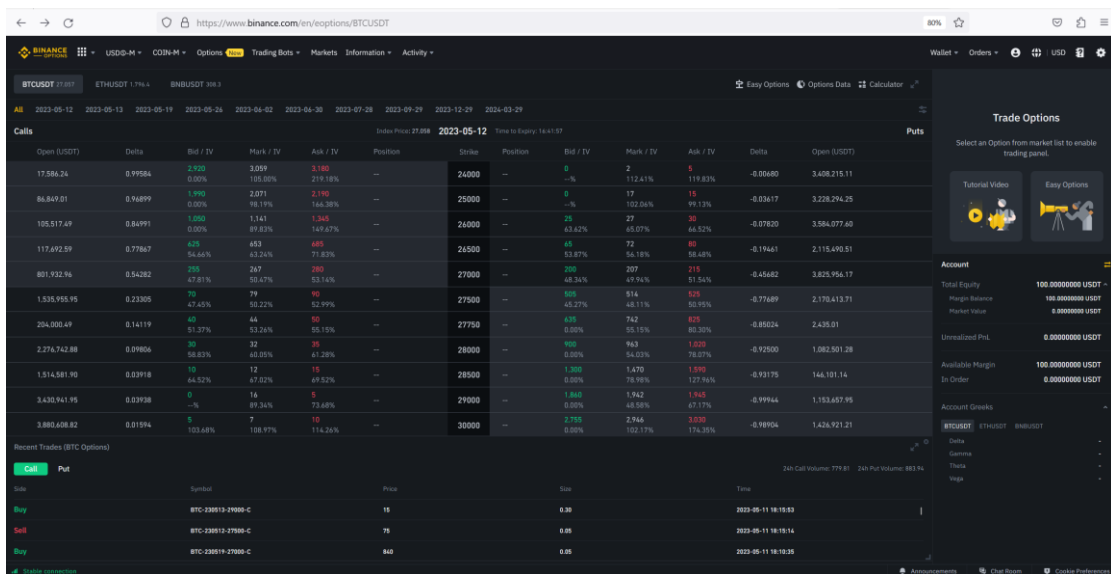
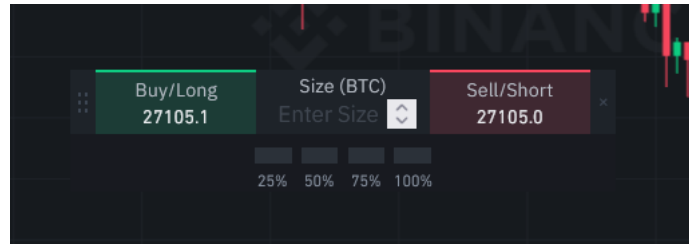


Рисунок 1.6 – Торгівля опціонами на криптобіржі Binance

1.7 Огляд ф'ючерсної позиції

Ф'ючерсна позиція – це фінансова позиція, яка виникає в результаті укладання ф'ючерсного контракту (рисунок 1.7). Ф'ючерсний контракт є договором між двома сторонами – покупцем та продавцем, що зобов'язує їх купити або продати криптовалюту за фіксовану ціну в майбутньому.

У ф'ючерській позиції покупець контракту (лонг) зобов'язується купити актив за встановленою ціною в майбутньому. Продавець контракту (шорт) зобов'язується продати актив за встановленою ціною в майбутньому.



*Рисунок 1.7 – Інтерфейс для швидкого відкриття позиції на криптобіржі
Binance*

Довга позиція (лонг) виникає, коли трейдер купує ф'ючерський контракт (рисунок 1.8). Трейдер, що утримує лонг, очікує підвищення ціни активу в майбутньому, щоб отримати прибуток після його продажу.

Коротка позиція (шорт) виникає, коли трейдер продає ф'ючерський контракт. Трейдер, що утримує (шорт), очікує зниження ціни активу в майбутньому, щоб отримати прибуток після його викупу.

Symbol	Size	Entry Price	Mark Price	Liq.Price	Margin Ratio	Margin	PNL(ROE %)
BTCUSDT Perpetual	0.001 BTC	27,147.30	27,160.34	--	0.01%	1.36 USDT (Cross)	+0.01 USDT (+1.17%)

Рисунок 1.8 – Огляд відкритої позиції на криптобіржі Binance

1.8 Огляд операцій для торгівлі з біржею

На криптобіржах ордери використовуються для розміщення та виконання торгових замовлень [12]. Ордер – це інструкція, яку трейдер надсилає біржі, щоб купити або продати певну криптовалюту за певну ціну.

Через інструменти для роботи з біржею, можемо виставляти різні види ордерів [14] для відкриття чи закриття позиції:

– LIMIT – ордер на купівлю/продаж за заданою ціною або краще; цей тип ордеру використовують для купівлі/продажу за фіксованою ціною (рисунок 1.9).

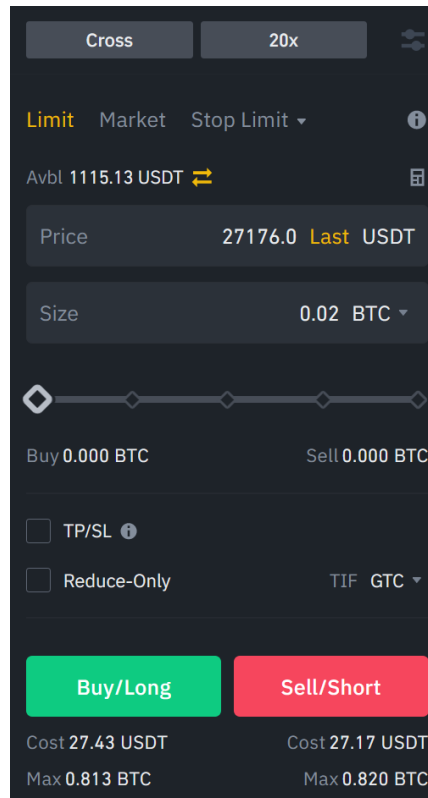


Рисунок 1.9 – Інтерфейс LIMIT ордеру на криптобіржі Binance

– MARKET – ордер на купівлю/продаж за поточною ринковою ціною (рисунок 1.10); цей тип ордеру використовують для швидкої операції купівлі/продажу.

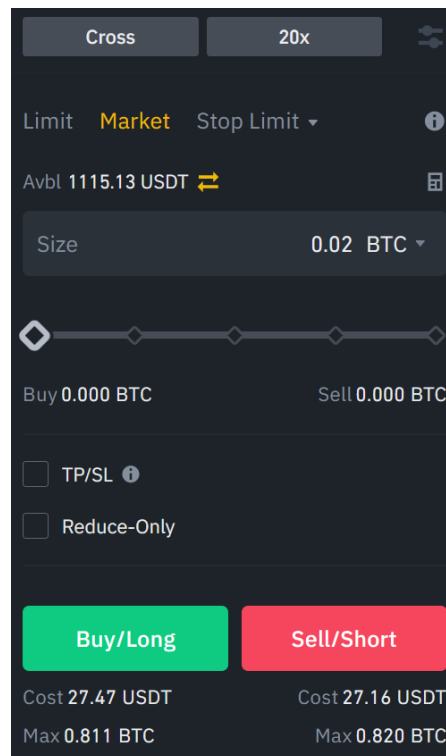


Рисунок 1.10 – Інтерфейс MARKET ордеру на криптобіржі Binance

– STOP-LOSS [13] – це протилежний ордер до ордеру входу, який автоматично виконується по ринковій ціні, коли збитки активів ростуть до заданого рівня (рисунок 1.11); цей тип ордеру використовують для захисту від зайвих збитків.

– TAKE-PROFIT [14] – це протилежний ордер до ордеру входу, який автоматично виконується по ринковій ціні, коли прибуток активів росте до заданого рівня; цей тип ордеру використовують для забезпечення прибутку, коли ринок рухається в бік відкритої позиції.

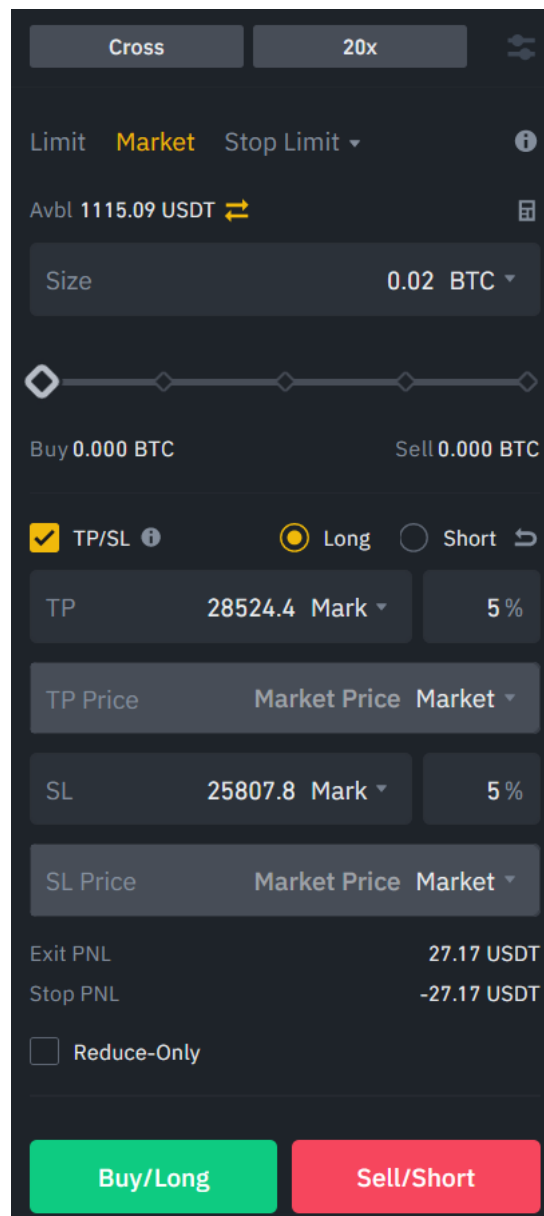


Рисунок 1.11 – Інтерфейс STOP-LOSS та TAKE-PROFIT для MARKET ордеру на криптобіржі Binance

– TRAILING STOP [15] – це модифікований STOP-LOSS ордер, який автоматично зміщується з рухом ціни; цей тип ордеру використовують для захисту від збитків та одночасно забезпечення можливості забезпечення прибутку у разі руху ціни в бік ціни ордеру входу; приклад інструменту вказано на рисунку 1.12.

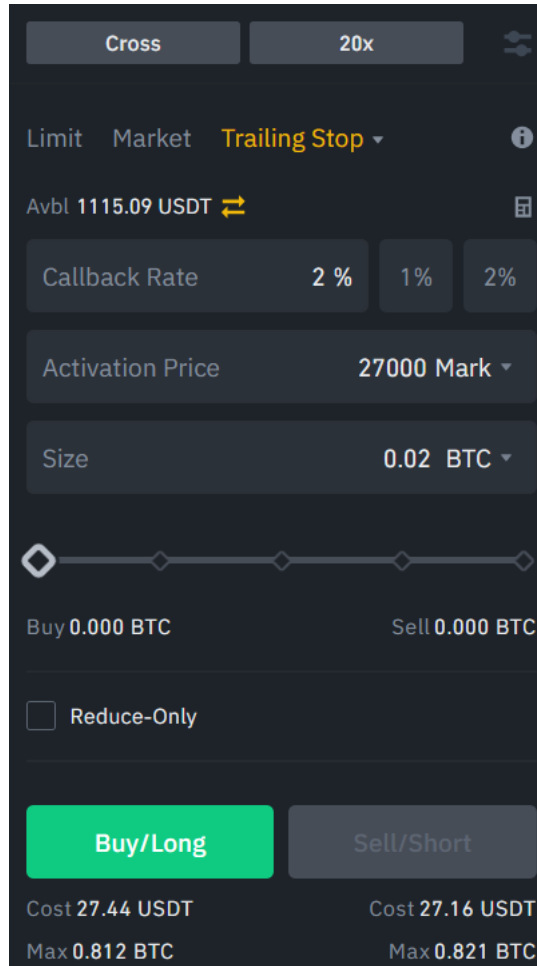


Рисунок 1.12 – Інтерфейс TRAILING STOP ордеру на криптобіржі Binance

1.9 Огляд ордербуку та трейдів

Ордербук (orderbook) [12] – список всіх активних купівельних та продажних ордерів для певного символу криптовалюти. Він відображає актуальну інформацію про ціни та обсяги (рисунок 1.13), за якими трейдери готові купувати або продавати криптовалюту. Кожен рядок містить ціну та обсяг, за яким трейдер хоче придбати чи продати криптовалюту

Ордери на купівлю (bids) виставлені трейдерами, які готові купити криптовалюту за фіксовану ціну. Ордери з більш високою ціною розташовуються вище, а ордери з меншою ціною розташовуються нижче.

А ордери на продаж (asks) виставлені трейдерами, які готові продати криптовалюту за фіксовану ціну. Ордери з меншою ціною розташовуються вище, а ордери з більш високою ціною розташовуються нижче.

Ордербук надає трейдерам важливу інформацію про поточну ринкову ліквідність та стан попиту та пропозиції для криптовалюти. Він дозволяє трейдерам оцінювати потенційні ціни для виконання угод і приймати рішення щодо покупки або продажу криптовалюти. Трейдери можуть переглядати ордербук, щоб побачити найкращі доступні ціни для покупки або продажу та визначити оптимальні рівні цін для своїх угод.

Price(USDT)	Size(BTC)	Sum(BTC)
26830.8	0.002	20.576
26830.7	0.003	20.574
26830.6	1.702	20.571
26830.5	0.002	18.869
26830.4	0.465	18.867
26830.3	0.019	18.402
26830.2	1.092	18.383
26830.1	17.291	17.291
26830.1 ↑	<u>26834.7</u>	
26830.0	21.585	21.585
26829.9	1.359	22.944
26829.8	0.038	22.982
26829.7	0.196	23.178
26829.6	0.338	23.516
26829.5	0.009	23.525
26829.4	0.002	23.527
26829.3	1.559	25.086

Рисунок 1.13 – Вигляд ордербуку на криптобіржі Binance

Коли ціна на продаж стає менше ціни на купівлю, або навпаки, то відбувається перетин ордерів – матчинг. Це процес, в якому ордери на продаж і

купівлю з ордербука зіставляються між собою для виконання угод. Коли ціни ордерів збігаються або перетинаються, відбувається матчинг та угода виконується. Внаслідок процесу виникають трейди.

Трейди на криптобіржі – це результат купівлі або продажу криптовалюти [16] між трейдерами на ринку. Кожен трейд представляє собою угоду, яка відбувається між двома сторонами: продавцем, який продає криптовалюту, і покупцем, який купує криптовалюту.

Трейди на криптобіржі відображаються в торговому журналі і включають такі дані, як ціна, обсяг, час (рисунок 1.14).

Trades		
Price(USDT)	Amount(BTC)	Time
27,009.4	0.275	00:48:10
27,009.5	0.027	00:48:10
27,009.4	0.056	00:48:09
27,009.5	14.341	00:48:09
27,009.0	0.254	00:48:08
27,009.0	0.287	00:48:07
27,009.1	0.815	00:48:07
27,009.0	0.954	00:48:06
27,009.1	2.346	00:48:06
27,009.1	0.065	00:48:05
27,009.0	1.137	00:48:04
27,009.0	0.096	00:48:04
27,009.0	0.042	00:48:04
27,009.1	0.289	00:48:03

Рисунок 1.14 – Вигляд трейдів на криптобіржі Binance

Зазвичай інформація про них оновлюється в режимі реального часу, так само, як і ордербук. Трейди дозволяють відстежувати останні зміни в цінах і обсягах торгів, і тому можуть бути використані для аналізу ринку, виявлення тенденцій та прийняття рішень щодо входу або виходу з позицій на основі актуальних торгових даних.

1.10 Огляд популярних стратегій торгівлі

Існує багато різних стратегій [17] для торгівлі криптовалютами, і їх використання залежить від різних факторів, таких як термін інвестування, обсяг інвестицій, цінові прогнози (рисунок 1.15) та інші.

Однак, описати декілька найпопулярніших стратегій можна у такий спосіб:

– денна торгівля (Day Trading) [18] – це стратегія, при якій трейдер купує та продає криптовалюту впродовж одного дня, зазвичай використовуючи технічний аналіз для здійснення операцій; такі трейдери зазвичай використовують великі позики та короткі позиції для отримання прибутку з невеликих рухів цін;

– свінг-трейдинг (Swing Trading) – це стратегія, при якій трейдер утримує позицію впродовж кількох днів до кількох тижнів [19]; в цьому випадку трейдери зазвичай зосереджуються на великих тенденціях ринку, використовуючи фундаментальний аналіз та технічний аналіз для здійснення операцій;

– хеджування (Hedging) [10] – це стратегія, при якій трейдери використовують інструменти, такі як ф'ючерсні контракти та опціони, для захисту від ризиків зміни ціни на ринку криптовалют; наприклад, трейдер може купити ф'ючерсний контракт на продаж на позицію еквівалентну його капіталу, щоб захистити свої інвестиції від можливого падіння цін;

– інвестування (HODL) – це стратегія, при якій трейдери вкладають кошти в криптовалюту на довгий термін, незалежно від короткострокових цінових рухів;

– скальпінг – стратегія скальпінгу передбачає виконання великої кількості швидких операцій [20] з метою отримання прибутку від невеликих рухів курсу; скальпери намагаються заробити на різниці між покупкою і продажем криптовалюту протягом дуже короткого часу, часто менше хвилини; ця стратегія вимагає від торговця високої концентрації і швидкості реакції, а також використання відповідного обладнання та програмного забезпечення.



Рисунок 1.15 – Приклад графічного аналізу крипторинку

Кожна зі стратегій має свої переваги і недоліки.

Доцільно розглянути комбінацію двох стратегій:

- хеджування – протилежна торгівля на двох ринках для захисту від зміни ціни;
- скальпінг – швидкий аналіз ціни з миттєвою реакцією для збільшення прибутків та підвищення надійності.

Основним прикладом такої комбінованої стратегії є використання арбітражу між криптовалютними біржами. Для цього трейдер купує криптовалюту на одній біржі за меншу ціну і продає її на іншій біржі за більш високу ціну. Різницю між курсами трейдер заробляє як прибуток.

Цю різницю називають спредом, а стратегію, що використовує такий спосіб торгівлі – торгівлею за спредом.

1.11 Торгівля за спредом

Для стратегії торгівлі за спредом [21] інвестор повинен використовувати дві торговельні пари з однаковою базовою цінністю, але різною фактичною ціною. Наприклад, пара BTC/USDT може мати іншу ціну, ніж пара BTC/BUSD на спільній для обох біржі. Ідея полягає в тому, щоб одночасно купувати одну торговельну пару та продавати іншу, використовуючи різницю (спред) (рисунок 1.16) між ними для отримання прибутку.

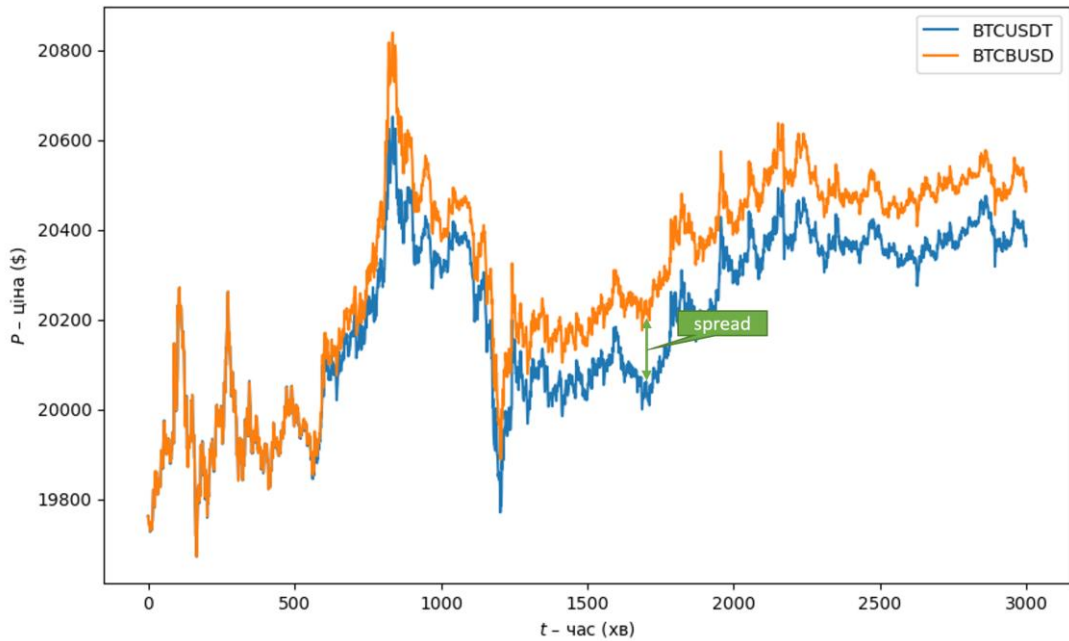


Рисунок 1.16 – Різниця між цінами, або спред

Прибуток виникає пізніше тоді, коли ціни двох пар збігаються, тобто коли спред зменшується. Схема торгівлі показана на рисунку 1.17.

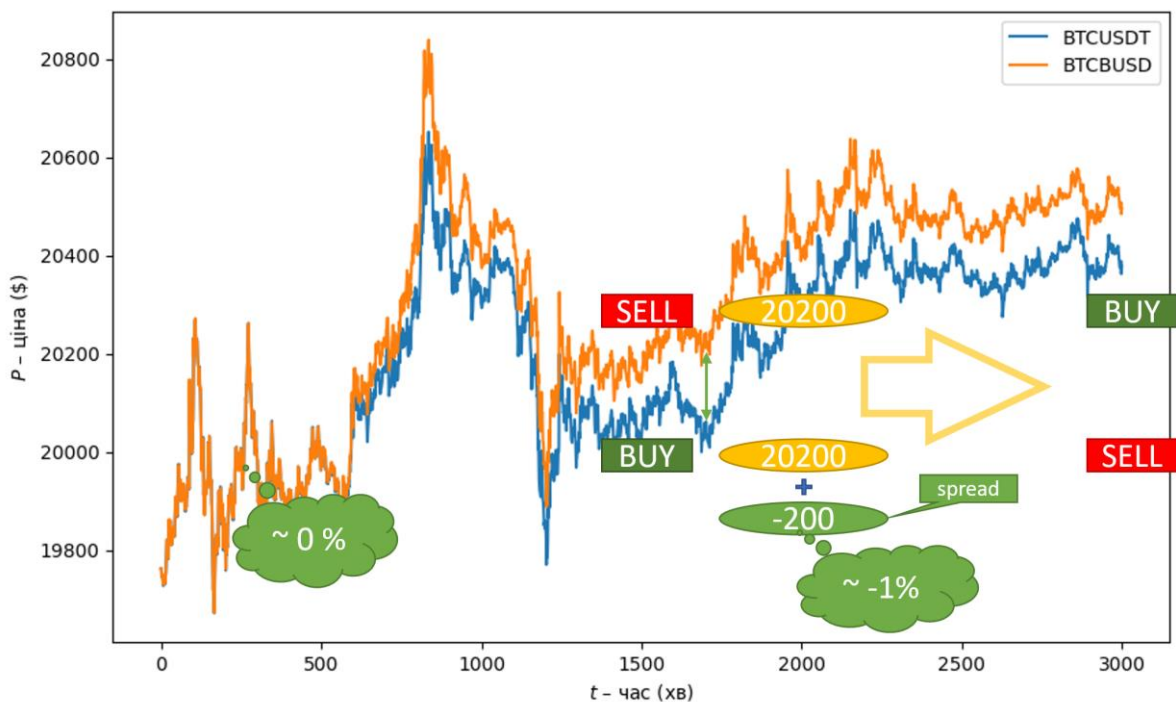


Рисунок 1.17 – Схема торгівлі за спредом

Наприклад, якщо курс BTC/BUSD становить \$20200, а BTC/USDT – \$20000, то інвестор може купити BTC/USDT за \$20000 і продати BTC/BUSD за \$20200, отримавши прибуток в розмірі \$200, не враховуючи комісію.

Використання показника спреда допомагає інвестору відслідковувати рух різниці між двома торговельними парами та аналізувати час входу та виходу з позицій. Якщо спред збільшується, то це може свідчити про можливість входу в позицію, а якщо спред зменшується, то можна виходити з позиції.

Оскільки торгівля за спредом базується на хеджуванні, ця стратегія дозволяє інвестору зменшити ризики та зберегти капітал в періоди невизначеності на ринку.

Для моментального хеджування позицій для торгівлі за спредом можна використовувати автоматичне програмне забезпечення, яке дозволяє здійснювати торговельні операції з використанням заздалегідь визначених параметрів та правил.

Для ефективного хеджування потрібно відстежувати спред між двома активами та виконувати операції з максимальною швидкістю. Оскільки швидкість виконання операцій [2] має критичне значення для успішної реалізації цієї стратегії, важливо мати доступ до швидкого Інтернет-з'єднання та розміщувати свої сервери якомога ближче до серверів біржі.

1.12 Алгоритм торгівлі за спредом

Для реалізації досліджуваної стратегії алгоритм дій має виконувати наступні функції:

- визначити спред між цінами купівлі та продажу відповідних торгових пар;
- надіслати LIMIT ордер на купівлю вище поточної ціни продажу з урахуванням спреда для першої сторони;
- надіслати LIMIT ордер на продаж нижче поточної ціни купівлі з урахуванням спреда для другої сторони;
- якщо один з LIMIT ордерів виконаний, відмінити інший та надіслати MARKET ордер на обернену операцію купівлі/продажу для іншої сторони.

Symbol	Size	Entry Price	Mark Price	Liq.Price	Margin Ratio	Margin	PNL(ROE %)	
BTCUSD Perpetual	20x 0.010 BTC	26,912.60	26,929.60	--	0.10%	13.46 USDT (Cross)	+0.17 USDT (+1.26%)	Market Limit 26928
BTCUSD Perpetual	50x -0.010 BTC	26,920.2	26,935.9	29,603.7	3.87%	5.39 BUSD (Cross)	-0.16 BUSD (-2.91%)	Market Limit 26940

*Рисунок 1.18 – Результат роботи алгоритму – дві позиції: купівля BTC/USDT,
продаж BTC/BUSD*

Застосування LIMIT ордеру дозволяє точно визначити вхідну ціну, а MARKET ордер дозволяє швидко виконати обернену операцію та відкрити іншу позицію. Проте, слід мати на увазі, що ця стратегія може бути вразлива до значних змін ринкових умов, що може призвести до збитків.

1.13 Математичний зміст спреду

За основу візьмемо те, що позиція буде відкриватись за допомогою LIMIT ордеру на продаж і буде хеджуватись MARKET ордером на купівлю. А закриття буде навпаки – LIMIT ордер на купівлю, а MARKET ордер на продаж.

Цінові тикери та ордербуки використаємо для обчислення спреду між криптовалютами парами.

Цінові тикери показують останню відому ціну, за якою здійснено угоду на ринку.

Для обчислення спреду між парами на основі цінових тикерів або ордербука, спочатку необхідно визначити ціни на кожен криптовалютну пару. Звичайно, ці ціни можуть змінюватися протягом дня, тому для точного визначення спреду необхідно здійснювати регулярне оновлення даних про ціну (рисунок 1.19).

Після отримання цін на кожен з пар, необхідно обчислити спред між цими парами, використовуючи формулу:

$$\text{Spread_Open} = ((\text{ASK_1} / \text{ASK_2}) - 1) * 100,$$

де Ask_1 – ціна продажу валюти на першій стороні, а Ask_2 – ціна купівлі валюти на другій, бо для закриття позиції ми використовуватимемо MARKET ордер на купівлю, який буде закривати пропозиції інших трейдерів на продаж.

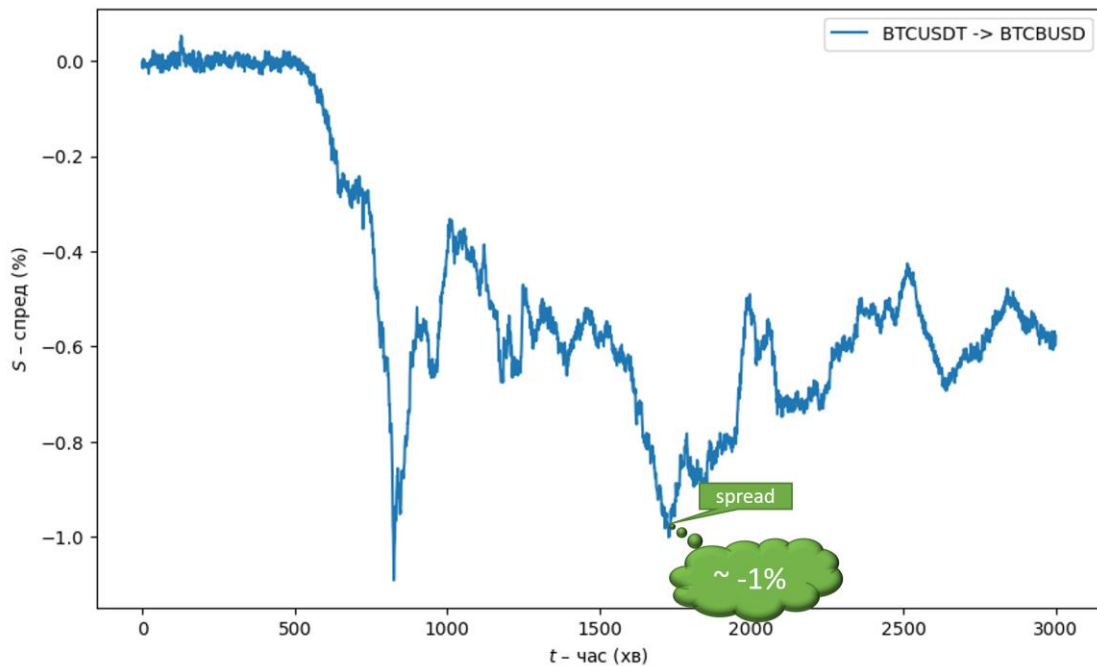


Рисунок 1.19 – Графік зміни спреду

Аналогічно для закриття:

$$\text{Spread_Close} = ((\text{BID}_1 / \text{BID}_2) - 1) * 100,$$

де BID_1 – ціна купівлі на першій стороні, а BID_2 – ціна продажу на другій.

1.14 Огляд сучасних інструментів для торгівлі

Існує багато сайтів та додатків, які дозволяють торгувати криптовалютами та використовувати різні стратегії. Розглянемо деякі з них:

- TradingView – це платформа для аналізу ринків та торгівлі на різних біржах, включаючи криптобіржі; вона містить інтерактивні графіки, індикатори та інші інструменти;
- Delta – це додаток для моніторингу криптовалютного портфеля та відстеження ринків;
- Coinigy – це платформа для торгівлі криптовалютами на багатьох біржах одночасно;
- 3Commas – це платформа для автоматизованої торгівлі криптовалютами, яка дозволяє створювати та налаштовувати торговельних ботів з використанням різних стратегій; 3Commas підтримує більше 20 криптобірж, включаючи Binance;

– TradeSanta – це платформа для автоматизованої торгівлі криптовалютами, яка дозволяє створювати та налаштовувати торгівельних ботів з використанням різних стратегій;

– Hummingbot – це відкрите програмне забезпечення для створення власних торгівельних ботів на криптобіржах; має вбудовані стратегії, а також може бути налаштований на виконання інших, створених користувачами;

– Kryll.io – це платформа для автоматизованої торгівлі криптовалютами, яка дозволяє створювати власні стратегії.

Для перевірки можливостей використання цих інструментів в стратегіях розглянемо кожен з них на можливість використання для торгівлі за спредом.

1.14.1 TradingView

TradingView – це онлайн-платформа для аналізу фінансових ринків та торгівлі різними інструментами, включаючи акції, ф'ючерси, криптовалюти та інші.

TradingView пропонує безкоштовний та платний доступ до віджетів для відображення графіків (рисунок 1.20), індикаторів, новин, календарів економічних подій та іншої інформації. Користувачі можуть налаштовувати графіки та індикатори за своїми потребами, створювати свої власні індикатори та скрипти.

Окрім цього, на платформі є можливість торгівлі через різні брокерські компанії, що підтримують інтеграцію з TradingView.

TradingView платформа дозволяє відслідковувати ціни різних торговельних пар на різних біржах і створювати відповідні технічні аналізи. Однак TradingView не є самостійною торговою платформою і не надає можливості для прямої торгівлі, тому торгівлю за спредом не можна виконати на TradingView без додаткових інструментів.



Рисунок 1.20 – Інтерфейс TradingView

Отже, для нашої задачі TradingView можна використовувати як засіб аналізу, але для виконання самої торгівлі потрібно використовувати інші платформи.

1.14.2 Delta

Delta – це платформа для відстежування та управління портфелем криптовалют. Вона доступна для використання на різних пристроях, включаючи веб-версію та мобільні додатки.

Основна функціональність Delta полягає у відстежуванні цін на криптовалюту та відображенні графіків цін в реальному часі. Користувачі можуть додавати свої активи до портфеля та відстежувати їх поточну вартість та зміни цін. Delta також надає інформацію про новини, пов'язані з ринком криптовалют, і рекомендації щодо оптимального розміщення замовлень на покупку та продаж.

Одним з основних переваг Delta є можливість зберігання більшості видів криптовалют в одному місці та автоматичне відслідковування їх цін. Також платформа пропонує інтеграцію з деякими криптобіржами, що дозволяє

користувачам відстежувати ціни (рисунок 1.21) та здійснювати управління портфелем безпосередньо з платформи.

Окрім того, Delta має функцію «додавання транзакцій», що дозволяє користувачам додавати вручну різні транзакції з криптовалютами та відстежувати свої інвестиції, враховуючи комісії та інші витрати.

Нажаль, на даний момент Delta не підтримує спеціалізовані інструменти для торгівлі за спредом.

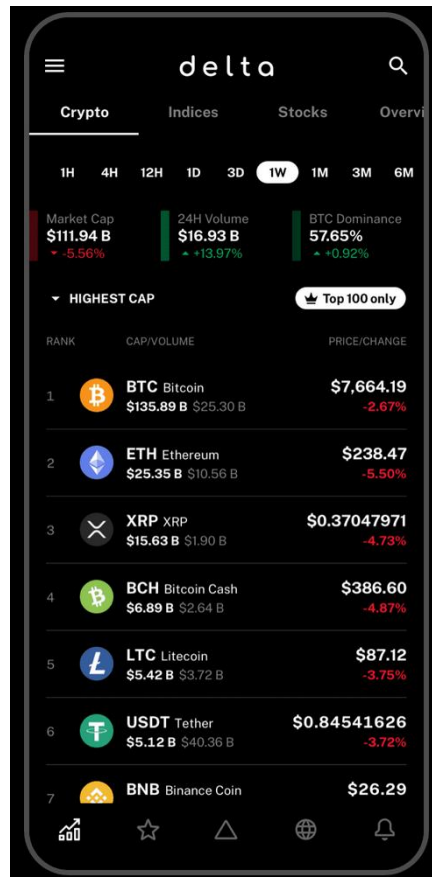


Рисунок 1.21 – Інтерфейс Delta

1.14.3 Coinigy

Coinigy – платформа для торгівлі криптовалютами, яка дозволяє торгувати на більш ніж 45 біржах з одного облікового запису. Крім того, вона забезпечує інструменти для аналізу ринку, портфелів та криптовалютних новин.

Основні функції Coinigy включають:

- глибини ринку та ордербуки з бірж;
- автоматичне відслідковування портфелю на кількох біржах;

– аналіз ринку, включаючи графіки (рисунок 1.22), індикатори та інші інструменти;

– налаштування та отримання сповіщень для цін, ордерів та новин;

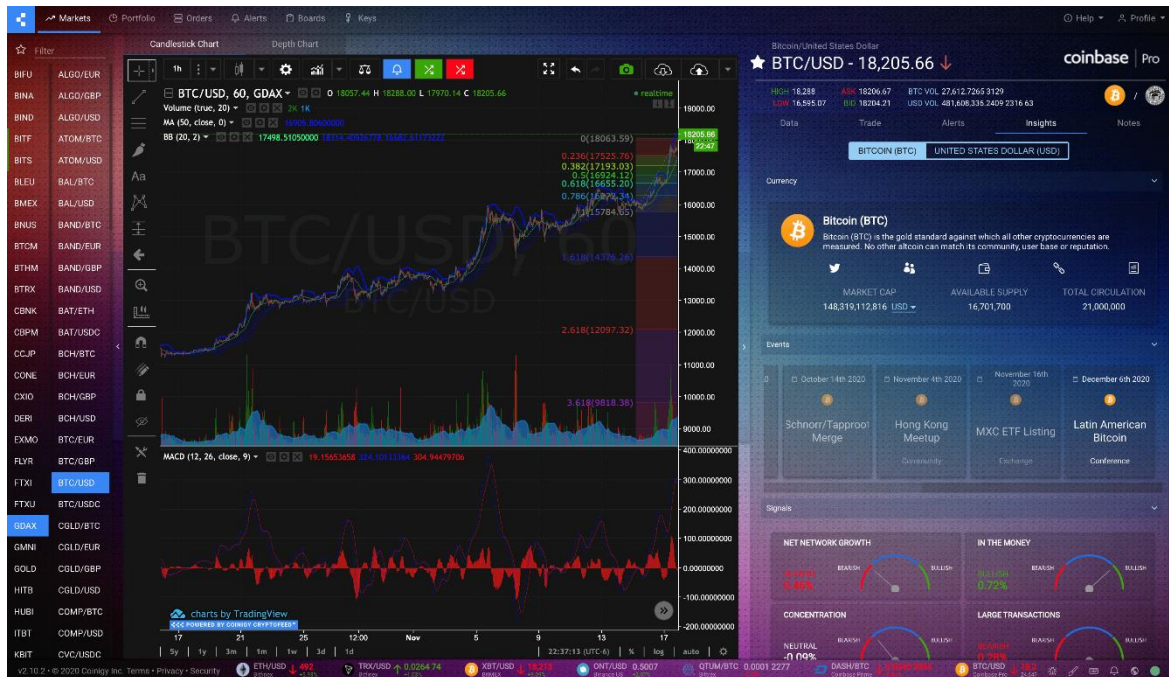


Рисунок 1.22 – Інтерфейс Coinigy

Для нашої задачі Coinigy можна використати, як інструмент для спостереження за цінами на декількох біржах.

1.14.4 3Commas

3Commas – це платформа, яка дозволяє трейдерам автоматизувати свої торговельні стратегії і вести торгівлю на криптовалютних біржах. 3Commas надає доступ до різноманітних інструментів для торгівлі криптовалютами, таких як:

– боти для автоматизації торгів забезпечують автоматизований аналіз ринку та здійснення торговельних операцій відповідно до налаштувань торговельної стратегії;

– панель управління портфелем дозволяє трейдерам відслідковувати свій поточний стан портфеля, включаючи загальний прибуток та збитки, графіки руху цін та іншу інформацію;

– сигнали для торгів надходять в режимі реального часу та містять важливу інформацію про можливі зміни на ринку, що дозволяє трейдерам приймати швидкі та обґрунтовані торговельні рішення.

Також 3Commas має вбудовану підтримку різних криптобірж, що дозволяє трейдерам використовувати платформу для торгівлі на будь-якій підтримуваній біржі.

3Commas дозволяє трейдерам налаштовувати свої торговельні боти (рисунок 1.23) для роботи з різними торговельними парами та виконувати торгівлю з використанням стратегій залежно від різниці цін між цими парами, але можливості для гнучкої торгівлі за спредом обмежені.

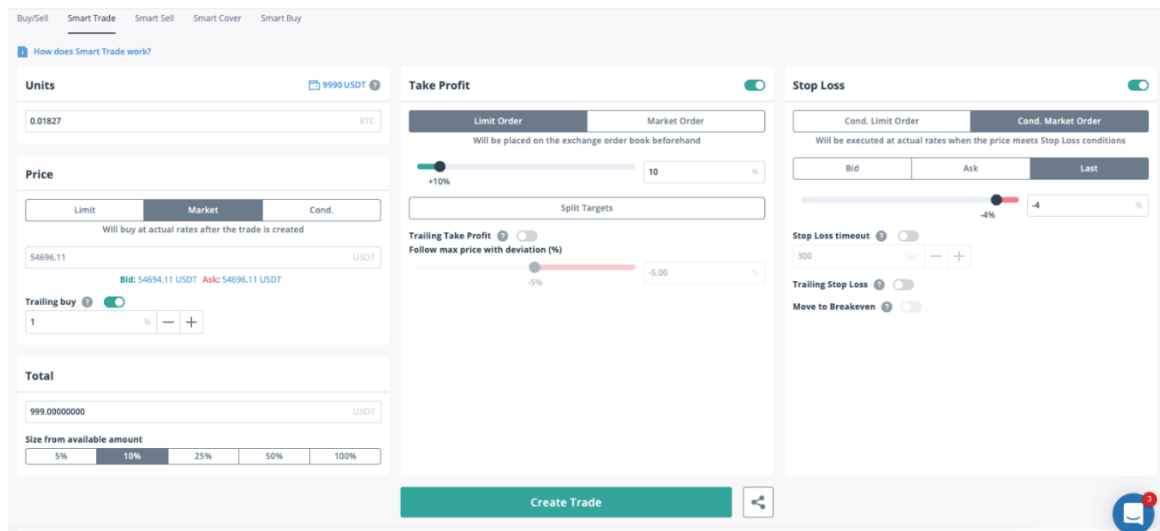


Рисунок 1.23 – Інтерфейс 3Commas

1.14.5 TradeSanta

TradeSanta – це платформа, яка надає інструменти для автоматизованої торгівлі на криптобіржах.

TradeSanta має інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко налаштувати свої торговельні боти (рисунок 1.24) і запустити їх у роботу. Платформа підтримує кілька криптобірж, включаючи Binance.

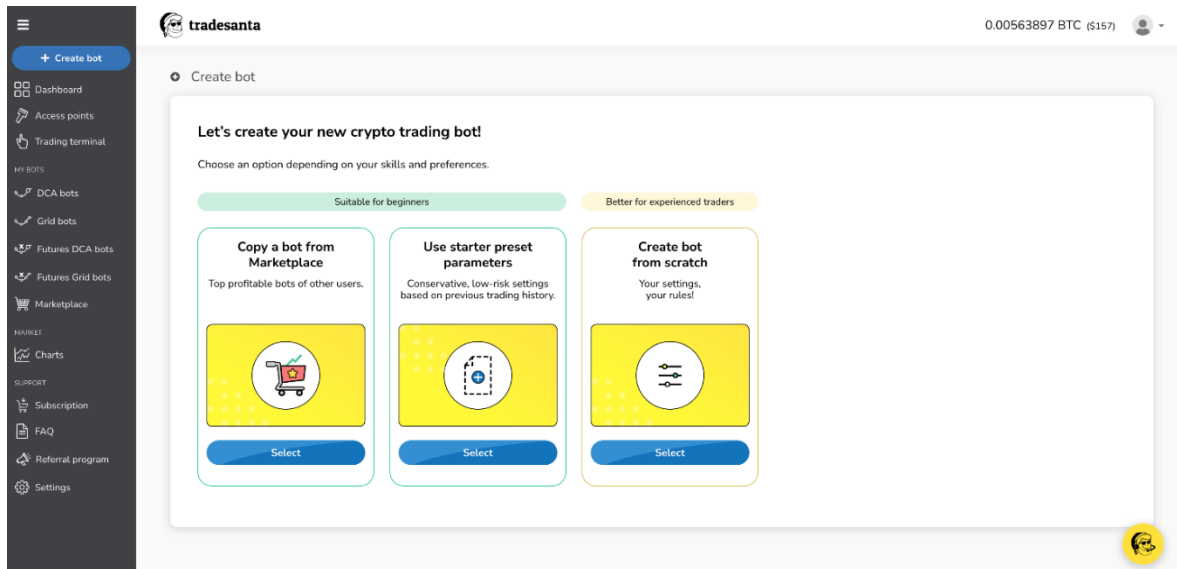


Рисунок 1.24 – Інтерфейс створення бота в TradeSanta

На TradeSanta можна налаштувати торгівельного бота (рисунок 1.25), що буде автоматично відкривати позиції на двох різних торгівельних парах і закривати їх при досягненні необхідного рівня прибутку або збитку.

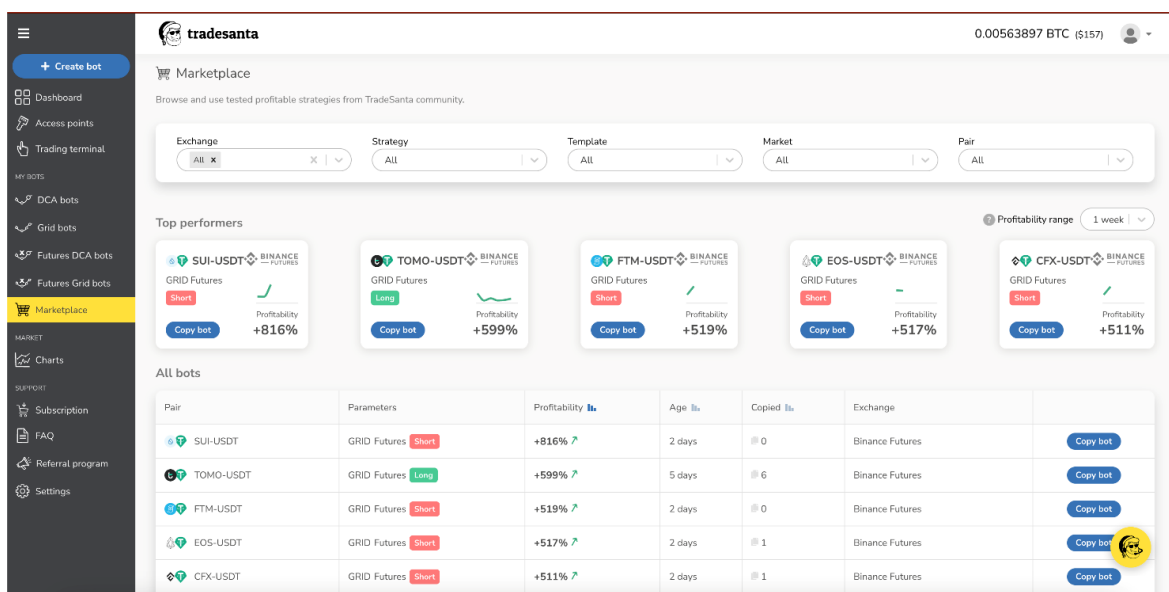


Рисунок 1.25 – Копіювання готового бота в TradeSanta

TradeSanta дозволяє користувачам встановлювати STOP LOSS і TAKE PROFIT ордери, що допомагає зменшити ризик у разі негативного руху цін на ринку.

1.14.6 Hummingbot

Hummingbot – це відкритий додаток для автоматичної торгівлі на різних криптобіржах. Додаток дає можливість торгувати криптовалютами з використанням алгоритмів, включаючи торгівлю за спредом.

Додаток дозволяє користувачам використовувати власні алгоритми для торгівлі на різних біржах і з різними парами, зберігаючи при цьому безпеку даних і фінансів. Користувачі можуть налаштувати свої торговельні стратегії і використовувати інтегровані.

Hummingbot має інтуїтивно зрозумілий інтерфейс користувача, що дозволяє налаштовувати параметри торговельної стратегії, такі як максимальна кількість позицій, максимальна кількість ордерів на позицію, мінімальний та максимальний розмір ордеру, мінімальний розмір кроку тощо.

Цей інструмент можна використовувати як на персональному комп'ютері, так і на сервері, що дозволяє користувачам отримувати максимальну швидкість торгівлі з мінімальними затримками. Крім того, Hummingbot підтримує зв'язок з більшістю провідних криптобірж.

1.14.7 Kryll.io

Kryll.io – це платформа для автоматизованої торгівлі криптовалютами, що дозволяє створювати та тестувати торговельні стратегії без програмування.

Основні можливості:

- візуальний конструктор стратегій дозволяє створювати торговельні стратегії за допомогою блоків без потреби в програмуванні;
- тестування стратегій дозволяє тестувати стратегії на історичних даних, щоб переконатися в їх ефективності перед реальною торгівлею;
- підтримка більше 15 бірж дозволяє торгувати на більш як 15 криптобіржах, включаючи Binance;
- різноманітність інструментів – платформа надає різноманітність інструментів для автоматизації торгівлі, такі як технічні індикатори, засоби управління ризиками та засоби для створення складних умовних ордерів;
- режим «папуга» дозволяє копіювати торговельні ідеї інших трейдерів та використовувати їх у своїй стратегії;

– віддалене управління – платформа має можливість віддаленого управління, що дозволяє контролювати та керувати торгівлею з будь-якої точки світу.

Krull.io надає широкі можливості для автоматизованої торгівлі на ринку криптовалют, що робить її привабливим вибором для трейдерів будь-якого рівня досвіду, але можливості для торгівлі за спредом дуже примітивні.

1.15 Огляд досліджуваної криптобіржі

Для роботи з криптовалютами активами усі розглянуті інструменти використовують інтеграцію з відкритими інтерфейсами криптобірж. Серед усіх інструментів найбільшу популярність має Binance. Це найбільша у світі криптобіржа за обсягом торгів і щоденним на березень 2023 року в більше ніж 50 мільярдів доларів США та 90 мільйонами клієнтів у всьому світі. Платформа зарекомендувала себе як надійний учасник криптопростору, де користувачі можуть купувати, продавати та зберігати свої цифрові активи, а також отримувати доступ до понад 350 існуючих криптовалют і тисяч торгових пар.

Binance має такі переваги порівняно з іншими:

- надійна та безпечна платформа з розширеними функціями безпеки, такими як двофакторна аутентифікація, захист від фішингу та захист від DDoS-атак;
- широкий вибір торгових пар, включаючи криптовалюти, токени та фіатні валюти;
- висока ліквідність та швидке виконання замовлень;
- низькі комісійні збори для торгівлі та виведення коштів;
- доступність різних інструментів для торгівлі (рисунок 1.26).

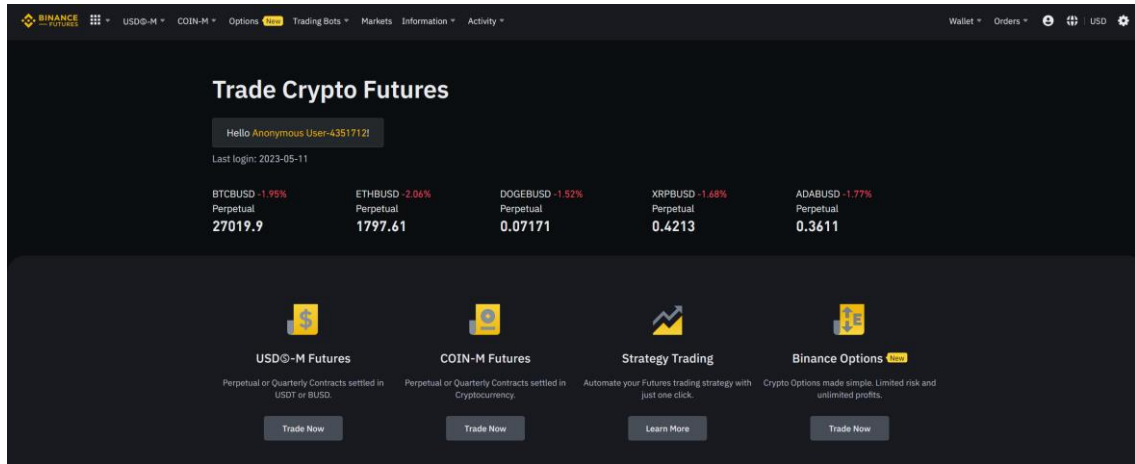


Рисунок 1.26 – Головна сторінка ф'ючерсної торгівлі на криптобіржі Binance

API Binance – це набір інтерфейсів (рисунок 1.27), які надають доступ до різних функціональних можливостей криптобіржі Binance, включаючи отримання даних про тикери, ордербук, обробку ордерів, доступ до балансу, позиції аккаунту та багато іншого.

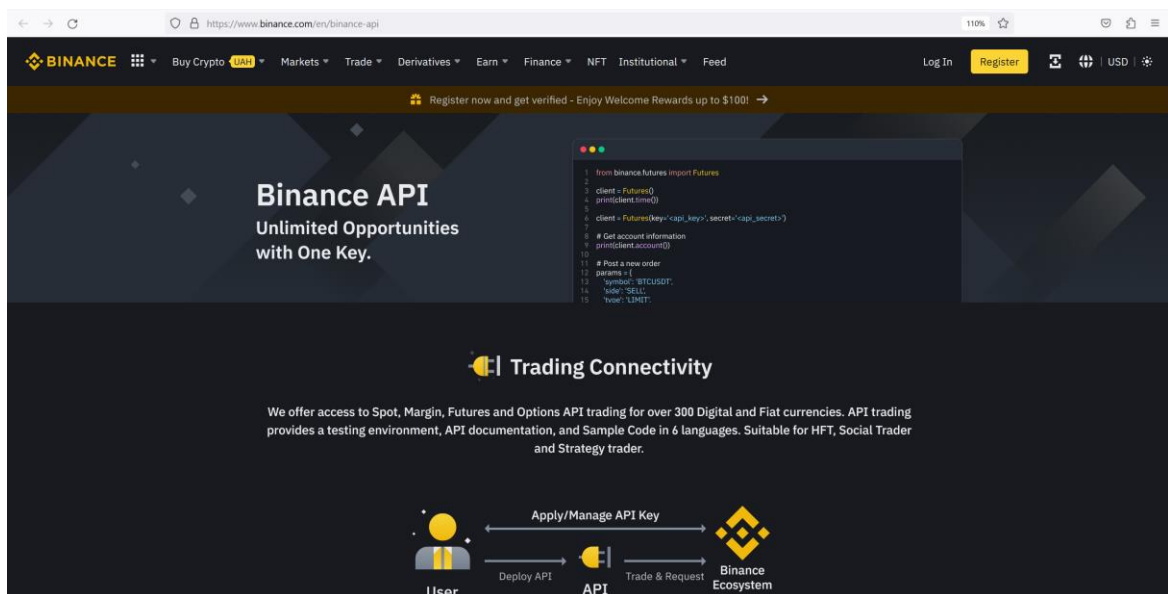


Рисунок 1.27 – Сторінка опису інтеграції з API на криптобіржі Binance

Документація Binance API надає детальну інформацію про доступні ендпоінти, параметри запиту, формати даних та інші важливі аспекти інтеграції з API.

1.16 Огляд інструменту для перевірки отриманих значень спреду на криптобіржі

Серед усіх досліджуваних інструментів Hummingbot має можливість використання параметр спреду для алгоритмічної торгівлі за обраною стратегією. Він вже має готове рішення, яке може автоматично виставляти ордери і відмінити їх коли потрібно з урахуванням навантаження. Також Hummingbot реагує на відкриту позицію і автоматично на іншій стороні виставляє протилежний ордер.

Для роботи цей інструмент потребує параметр спреду, який повинен прописати користувач. Для обрахунку цього значення розглянемо систему збору та обробки даних з криптобіржі Binance в наступному розділі.

Висновки до розділу

Розглянуто процеси, які використовуються для торгівлі на крипторинку. Переглянуті джерела, які описують сучасні реалізації інтеграцій торгових алгоритмів з криптобіржами.

Оглянувши сучасні криптобіржі, для дослідження обрано Binance. Проаналізовано роботу ф'ючерсних контрактів та типи ордерів, які можна використати: LIMIT (фіксована ціна) та MARKET (ринкова ціна) ордери. Також досліджені ризики роботи з капіталом на цій біржі.

Розглянуто декілька основних стратегій для торгівлі. Обрана торгівля за спредом. Перевірено ефективність її використання з урахуванням ризиків та можливостей до автоматизації. Для дослідження обраний наступний варіант торгівлі – ф'ючерсний контракт проти іншого ф'ючерсний контракту за однаковою базовою ціною.

Проаналізовано стратегію та виокремлені параметри для роботи з нею. Описано параметр спреду, який описує відсоткове відношення ціни на одній стороні проти іншої. Для покращення результату потрібно розглянути методи прогнозування та обрати один з них.

2 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ ДЛЯ ЗБОРУ, АНАЛІЗУ, ЗБЕРІГАННЯ ТА ВІДОБРАЖЕННЯ ДАНИХ З КРИПТОБІРЖ

2.1 Способи збору даних з криптобірж

Щоб інтегрувати стратегію з криптобіржею потрібно визначити, якими засобами будемо взаємодіяти з нею. Розглянемо поширені способи взаємодії з криптобіржами:

– REST API – стандарт, який використовується на усіх популярних біржах для отримання інформації і надсилання команд створення ордерів;

– WebSockets [23] – двосторонню зв'язок між клієнтом і сервером (рисунок 2.1), що дозволяє отримувати оновлення ринкових та клієнтських даних в режимі реальному часі;

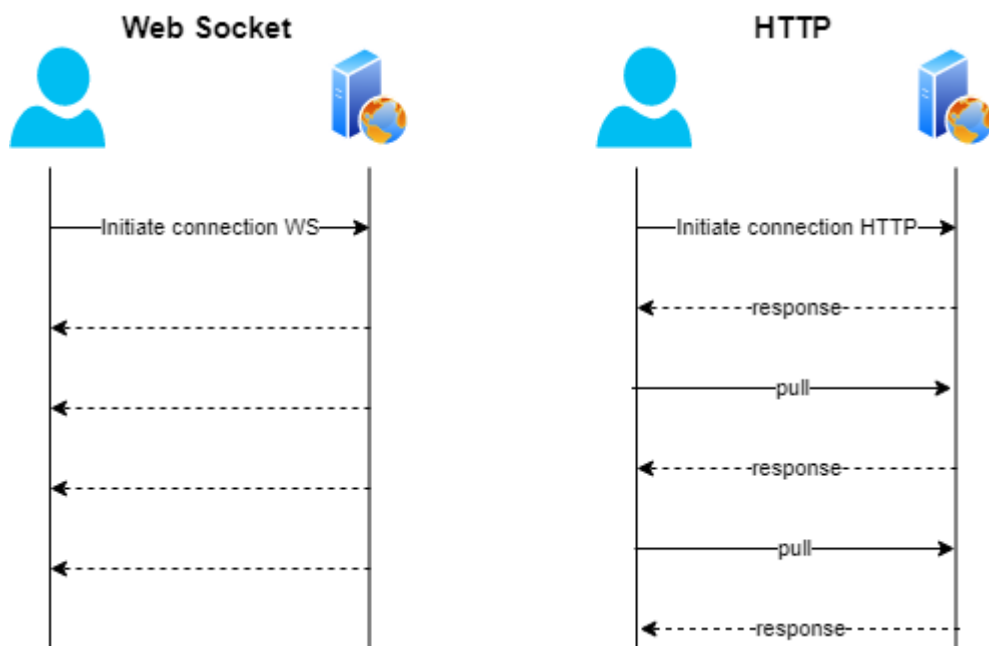


Рисунок 2.1 – Оновлення даних за допомогою WebSockets та REST

– спеціалізоване програмне забезпечення – деякі криптобіржі надають власне програмне забезпечення, яке дозволяє вам підключитися та взаємодіяти з біржею безпосередньо через своє спеціалізоване ПЗ; це може бути у вигляді торгових терміналів або платформ, які надають додаткові можливості для автоматизованого торгівельного аналізу та виконання стратегій;

– FIX протокол – Financial Information eXchange (рисунок 2.2) є стандартним протоколом, який використовується у фінансовій індустрії для передачі даних та виконання торгових операцій; це стандартний протокол обміну фінансовою інформацією між учасниками фінансових ринків, такими як брокери, банки та біржі. FIX протокол був розроблений з метою стандартизації та спрощення комунікації між різними фінансовими системами.

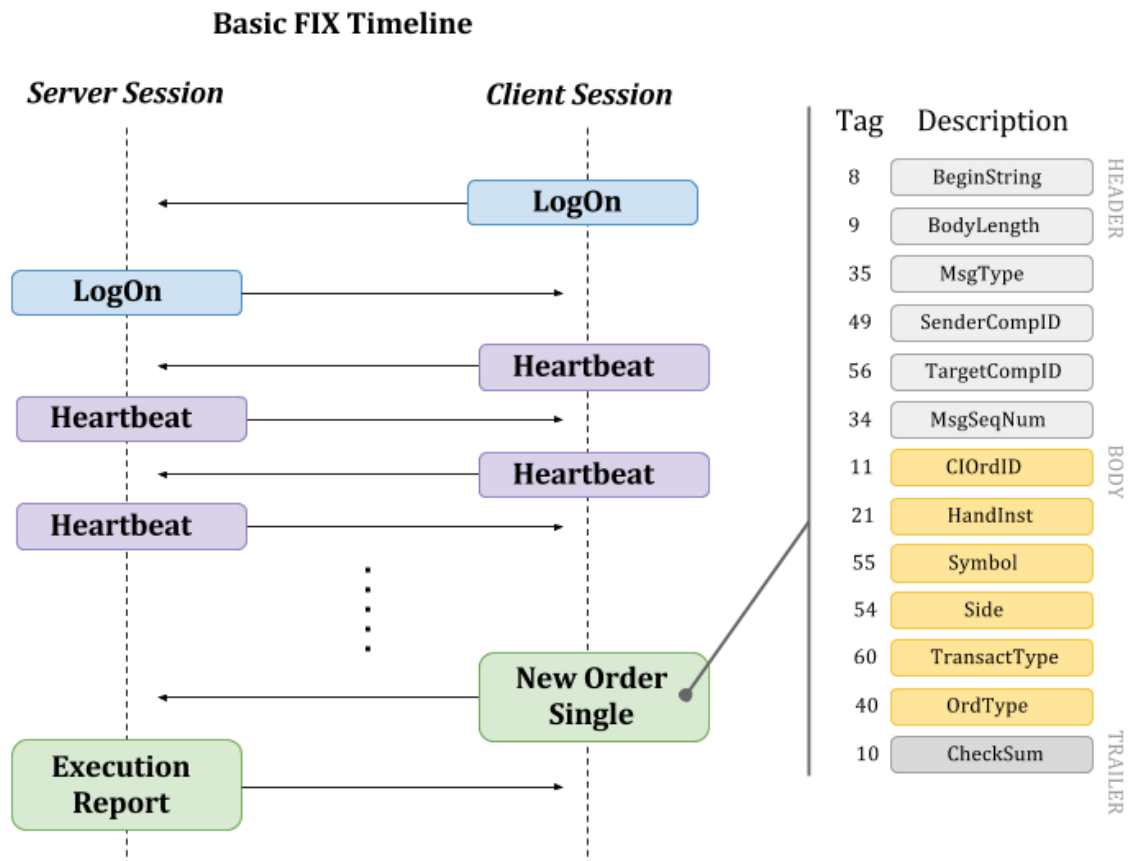


Рисунок 2.2 – Авторизація та оновлення даних за допомогою FIX протоколу

Кожен з цих способів має свої переваги та недоліки, тому в залежності від обраної задачі обирається той, чи інший інструмент.

2.2 Огляд видів даних з криптобірж

Розглянемо популярні види даних з криптобірж, які можна отримати через REST або WebSockets:

– тикери – інформація про поточні ціни та стан ринку для певного криптовалютного активу;

- ордербук – виставлені трейдерами ордери на купівлю та продаж;
- трейди – історичні дані про здійснені угоди купівлі або продажу криптовалютних активів.

Усі ці дані криптобіржа надсилає у вигляді структурованої інформації – форматі JSON.

2.2.1 Формат даних тикерів

Структура тикеру:

- символ – унікальний ідентифікатор криптовалютного активу, для якого надається інформація; наприклад, символом може бути пара BTC/USDT для обміну Bitcoin на криптовалютний долар від компанії Tether;

- найкраща ціна продажу (BestAsk) – найкраща доступна ціна, за якою можна продати актив на даному моменті;

- найкраща ціна покупки (BestBid) – найкраща доступна ціна, за якою можна купити актив на даному моменті;

- час зміни однієї з кращих цін.

Приклад формату REST API для криптобіржі Binance:

```
{
  "symbol": "BTCUSDT",
  "bidPrice": "26459.30",
  "bidQty": "0.274",
  "askPrice": "26459.40",
  "askQty": "7.168",
  "time": 1683901642630
}
```

Приклад формату WebSockets для криптобіржі Binance:

```
{
  "e": "bookTicker",
  "T": 1683901642630,
  "s": "BTCUSDT",
  "b": "26459.30",
  "B": "0.274",
```

```
"a": "26459.40",
"A": "7.168"
}
```

2.2.2 Формат даних ордербуку

Ордербук складається з двох основних частин, які сформовані з двох типів ордері на купівлю та продаж. Кожна з цих частин містить список цін і відповідних обсягів активів, які доступні для продажу або купівлі відповідно.

Структура кожного запису ордербуку:

- ціна, за якою учасники ринку готові купувати або продавати актив;
- обсяг – кількість активу, яку учасники ринку бажають купити або продати за вказаною ціною.

Приклад формату REST API для криптобіржі Binance:

```
{
  "lastUpdateId": 2838558205292,
  "E": 1683928859391,
  "T": 1683928859379,
  "bids": [
    ["26742.40", 7.910],
    ["26742.30", "0.002"],
    ["26742.20", 0.002]
  ],
  "asks": [
    ["26742.50", "5.562"],
    ["26742.60", "0.003"],
    ["26742.70", "0.003"],
  ]
}
```

Приклад формату WebSockets для криптобіржі Binance:

```
{
  "e": " depthUpdate ",
  "E": 1683928859391,
```

```

"T": 1683928859379,
"s": " BTCUSDT ",
"b": [
  ["26742.40", 7.910],
  ["26742.30", "0.002"],
  ["26742.20", 0.002]
],
"a": [
  ["26742.50", "5.562"],
  ["26742.60", "0.003"],
  ["26742.70", "0.003"],
]
}

```

2.2.3 Формат даних трейдів

Структура трейдів:

- ціна, за яку була здійснена угода;
- обсяг – кількість активу, що була куплена або продана в рамках ордеру;
- сторона – купівля чи продаж;
- час виконання угоди.

Приклад формату REST API для криптобіржі Binance:

```

{
  "id": 3707516882,
  "price": "26675.10",
  "qty": "0.030",
  "quoteQty": "800.25",
  "time": 1683932389885,
  "isBuyerMaker": false
}

```

Приклад формату WebSockets для криптобіржі Binance:

```

{
  "e": " trade ",

```

```

"E": 1683932389885,
"s": " BTCUSDT ",
"t": 3707516882,
"p": "26675.10",
"q": "0.030",
"T": 1683932389885
}

```

2.3 Способи покращення обробки даних з криптобірж

Обробка великої кількості даних з криптобірж може бути викликом через їх великий обсяг і високу швидкість змін. Існує кілька підходів, які можна використовувати для ефективної обробки таких даних:

- потокова обробка – спосіб обробки неперервного потоку даних в реальному часі; дозволяє обробляти, аналізувати і реагувати на дані, які надходять у потоковому режимі без затримок;

- паралельна обробка – використання технологій паралельної обробки даних може покращити швидкість і продуктивність обробки; потрібно розбивати потік даних на менші частини та обробляти його паралельно в різних обчислювальних ресурсах;

- оптимізація алгоритмів може значно зменшити час обробки; використання індексів, хешування, оптимізованих алгоритмів пошуку та фільтрації можуть допомогти покращити продуктивність системи;

- агрегація – велика кількість даних може бути скомпресована або агрегована до більш малих наборів даних для спрощення обробки;

- масштабування – при обробці великої кількості даних може виникнути необхідність в масштабуванні обчислювальних ресурсів;

- моніторинг – важливо систематично перевіряти продуктивність і шукати можливості для оптимізації; використання інструментів моніторингу продуктивності, таких як системи трасування запитів, допоможе виявити проблемні місця і покращити продуктивність обробки даних;

– резервне копіювання та відновлення – збір і обробка великої кількості даних з криптобірж вимагають дбайливого підходу до резервного копіювання і забезпечення можливості відновлення даних; регулярне створення резервних копій і зберігання їх в безпечному місці дозволить уникнути втрати даних в разі непередбачуваних ситуацій.

2.4 Паралельна потокова обробка

Основна ідея потокової обробки полягає в тому, що дані обробляються порціями або подіями, які поступають в потік, замість обробки всього набору даних. Це дозволяє в реальному часі отримувати результати, виконувати аналітику, приймати рішення і вживати заходів на основі нових даних, що надходять.

Потокова обробка складається з таких етапів:

– підписка на потік даних – отримання потоку даних з джерела, такого як криптобіржа або інший джерело даних; це може виконуватися за допомогою публічного API криптобіржі або спеціального інструменту для отримання поточних даних;

– обробка подій – кожна подія в потоці даних обробляється окремо; може включати фільтрацію, перетворення та аналіз подій, щоб отримати необхідну інформацію;

– аналітика та обчислення – у поточній обробці можна виконувати аналітику, обчислення та агрегацію інформації з потоку даних; може включати статистичний аналіз, виявлення аномалій, побудову моделей машинного навчання або будь-які інші операції, що вимагають обробки даних у режимі реального часу;

– реагування та вивід результатів – після обробки кожної події у потоці даних можна приймати рішення та вживати відповідних заходів на основі отриманих результатів; може включати виведення результатів в інструмент візуалізації, сповіщення адміністратора про виявлені проблеми або автоматичне виконання дій, таких як розміщення торговельних замовлень на криптобіржі.

Паралелізм в потоковій обробці використовується для одночасного виконання різних обчислень або операцій над подіями в потоці даних. Це дозволяє поліпшити продуктивність та ефективність обробки великого обсягу даних.

Існує декілька рівнів паралелізму, які можуть бути застосовані в потоковій обробці:

- рівень подій – події в потоці даних можуть бути оброблені паралельно; замість послідовної обробки подій, вони можуть бути розподілені між різними обробниками або робочими потоками для одночасного виконання; кожен обробник або потік виконує свою задачу над окремими подіями, що дозволяє збільшити швидкість обробки;

- рівень операцій – в рамках обробки однієї події можуть бути виконані різні операції паралельно; наприклад, аналіз даних, виконання обчислень, розрахунок статистики або моделей машинного навчання можуть бути розподілені між різними обчислювальними ресурсами або процесами для швидшої обробки;

- рівень задач – різні задачі в потоковій обробці можуть бути розподілені між різними вузлами або обчислювальними системами; наприклад, одна система може бути відповідальна за збір даних, інша – за обробку, а третя – за аналітику; кожна задача виконується паралельно на своєму ресурсі, що дозволяє забезпечити розділення обов'язків і прискорити процес обробки даних.

Для реалізації паралелізму в потоковій обробці можна використовувати різні технології:

- розподілені обчислення – використання розподілених обчислень дозволяє розподіляти обробку даних між різними вузлами або комп'ютерами;

- багатопоточність – використання багатопоточності дозволяє виконувати різні операції паралельно на різних потоках в рамках одного процесу або системи; це може бути досягнуто за допомогою мов програмування, які підтримують багатопоточність, де можна створювати багато потоків для виконання операцій над подіями в потоці даних;

– розподілені системи обробки потоків даних – існують спеціалізовані системи, які призначені для обробки потоків даних та надання можливостей паралельної обробки; вони можуть використовуватися для збирання, обробки та аналізу поточкових даних з підтримкою паралельних операцій та розподіленої обробки.

Застосування паралелізму в потоковій обробці дозволяє виконувати більше операцій одночасно та підвищує продуктивність обробки даних. Однак варто враховувати, що ефективне використання паралелізму також вимагає управління конкуренцією, синхронізації даних та уникнення гонок ресурсів.

2.5 Агрегація та перетворення даних

Використання агрегації та перетворення даних в контексті криптобірж означає об'єднання і зведення великої кількості даних в компактні, корисні та легкі для аналізу формати. Агрегація даних є важливою складовою процесу обробки і аналізу даних з криптобірж та надає наступні переваги:

– зведення даних – об'єднання розсіяних даних з різних джерел і форматів в одне ціле;

– скорочення обсягу даних – зменшення обсяг великої кількості даних шляхом відбору ключових метрик або показників; замість роботи з повним набором даних, можна зробити агрегацію даних за певний період (наприклад, годину, день або місяць) та обчислити сумарні значення, середні, мінімальні або максимальні значення;

– створення звітів та аналітичних даних – створення звіти та аналітичні дані, які дозволяють зрозуміти ключові тенденції, статистику та поведінку ринку криптовалют; це може включати аналіз торгових обсягів, цін, волатильності, ліквідності та інших факторів;

– покращення прийняття рішень – агреговані дані дозволяють зробити краще та більш обґрунтовані рішення на основі підставних фактів та аналітики; аналіз агрегованих даних може допомогти виявити тренди, шаблони та кореляції між різними показниками на криптобіржі; наприклад, агрегація даних може

допомогти виявити залежність між ціною криптовалюти і торговим обсягом або виявити певні шаблони та сезонність в коливаннях ціни;

- оптимізація продуктивності – покращення продуктивність обробки та аналізу даних; замість роботи з великими обсягами даних, можна працювати зі сконденсованими агрегованими даними, що дозволяє прискорити час виконання запитів та знизити вимоги до обчислювальних ресурсів;

- поліпшення візуалізації даних – створення зрозумілих та інформативних візуалізації даних; зведені дані можуть бути представлені у вигляді графіків, діаграм та інших візуальних елементів, що сприяють легшому розумінню та інтерпретації даних.

Загалом, агрегація та перетворення даних дозволяють витягти цінну інформацію з великого обсягу даних з криптобірж і зробити їх більш зрозумілими та корисними для подальшого аналізу, прийняття рішень та розробки стратегій на основі даних ринку криптовалют.

2.6 Використання брокерів повідомлень

Використання брокерів повідомлень [23] є ефективним підходом для розробки системи збору та збереження даних з криптобірж. Брокери повідомлень є посередниками між джерелами даних, системою обробки та збереження, і вони забезпечують надійну та ефективну передачу даних.

Переваги використання брокерів повідомлень:

- надійність та гарантована доставка – забезпечують надійну доставку даних між джерелами даних та системою збереження; вони використовують протоколи забезпечення надійності, такі як підтвердження отримання, повторна передача та механізми відновлення з'єднання, щоб гарантувати, що жодні дані не будуть втрачені під час передачі;

- масштабованість – можуть обробляти велику кількість даних та взаємодіяти з багатьма джерелами даних одночасно; це дозволяє розширювати систему збору даних з криптобірж, надаючи можливість обробляти більше потоків даних з різних джерел паралельно;

– фільтрація та маршрутизація – можуть фільтрувати та маршрутизувати дані відповідно до заданих правил; вони дозволяють вибирати лише необхідні дані для збору та передавати їх до відповідних систем обробки; це допомагає зменшити навантаження на систему та ефективно використовувати ресурси;

– розширюваність – є можливість легко додавати нові джерела даних або розширювати функціональність системи збору даних; можна налаштувати брокер повідомлень для отримання даних з нових джерел або розширити його функціональність шляхом додавання нових правил обробки даних; це дозволяє гнучко адаптувати систему збору даних до змінних вимог і розширювати її можливості;

– інтеграція з іншими системами – підтримують різні протоколи та інтерфейси, що дозволяє легко інтегрувати їх з іншими системами;

– резервне копіювання та відновлення – можуть забезпечити механізми резервного копіювання та відновлення даних; це важливо для забезпечення безпеки та надійності даних, особливо в разі відмови системи або випадків втрати даних.

Використання брокерів повідомлень дозволяє створити розподілену та надійну систему збору та збереження даних з криптобірж. Вони спрощують передачу даних, забезпечують гнучкість, масштабованість та інтеграцію з іншими системами, а також допомагають забезпечити безпеку та надійність даних.

2.7 Забезпечення надійності збору даних

Для забезпечення перевірки надійності даних отриманих з потоку та можливості повторної підключення у разі проблем, можна використовувати наступні технології:

– засоби контролю цілісності даних [24] – контрольні хеш-суми або підписи для перевірки цілісності отриманих даних; кожен блок даних може містити контрольну суму або підпис, який може бути перевірений, щоб впевнитися, що дані не були змінені під час передачі;

– механізми перевірки доставки – підтвердження отримання, зворотний зв'язок або використання протоколів, які гарантують доставку повідомлень, такі як TSP з підтвердженням доставки;

– повторна передача даних – механізм для повторної передачі даних у випадку втрати з'єднання або недоступності; при виявленні проблеми з'єднання або невдалої передачі даних, система повинна здатися на нове з'єднання та повторно надіслати відсутні дані;

– механізми перезавантаження – коли є проблеми з недоступністю компонентів чи інструментів системи; це може бути автоматичне перезавантаження або спроби підключення до альтернативних серверів;

– моніторинг та логування – систему моніторингу, яка буде слідкувати за станом з'єднання та обробки даних; це допоможе виявити проблеми та недоліки у реальному часі, а також зберегти журнали подій для подальшого аналізу; важливо зберігати достатньо інформації про стан з'єднання, передачу даних та помилки, щоб виявити проблеми та прийняти відповідні заходи;

– резервне зберігання даних – зберігання отриманих даних в проміжному сховищі або журналах, щоб мати можливість відновлення в разі непередбачених ситуацій; це дозволить зберегти дані, навіть якщо виникне відмова або проблема з обробкою;

– механізми розподіленої обробки – розподілені системи обробки даних для підтримки надійності та масштабованості; застосування розподіленої обробки дозволить розподілити навантаження та обробку даних між багатьма вузлами, що забезпечить зменшення ризику втрати даних та можливість продовжувати роботу навіть при відмові окремих компонентів.

Для забезпечення надійності даних отриманих з потоку і можливості повторного підключення у випадку проблем, важливо використовувати комбінацію засобів контролю цілісності, механізмів перевірки доставки, повторної передачі даних, моніторингу та резервного зберігання. Такі підходи забезпечать надійну та стійку систему збору та збереження даних з криптобірж.

2.8 Особливості роботи брокерів повідомлень

Брокер повідомлень є ключовим компонентом архітектури мікросервісних систем та мереж. Вони забезпечують передачу повідомлень між різними компонентами системи, дозволяючи їм взаємодіяти незалежно від їх фізичного розташування або конкретних протоколів комунікації.

Принципи роботи брокерів повідомлень:

– публікація та передача повідомлень – брокери повідомлень приймають повідомлення від відправника, який їх публікує на певному каналі або темі; після цього брокер розсилає це повідомлення всім підписникам, які підписалися на цей канал або тему (рисунки 2.3, 2.4);



Рисунок 2.3 – Публікація повідомлень за допомогою брокера повідомлень *RabbitMQ*

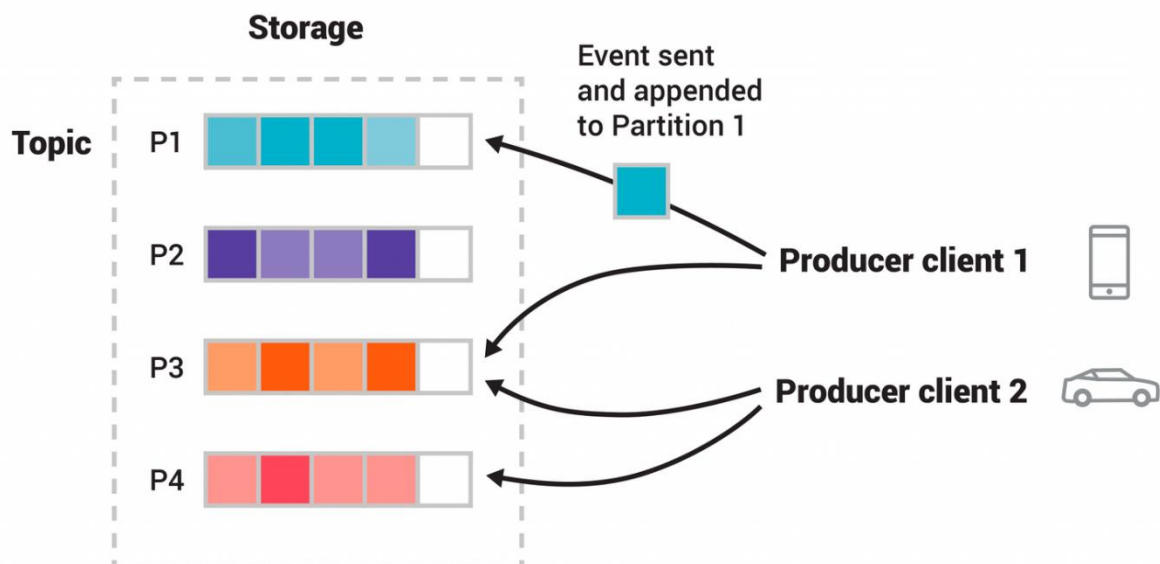


Рисунок 2.4 – Публікація повідомлень за допомогою брокера повідомлень *Kafka*

– підписка на повідомлення – компоненти системи можуть підписуватися на певні канали або теми (рисунок 2.5); брокер повідомлень утримує список підписників для кожного каналу та відстежує їхні підписки;

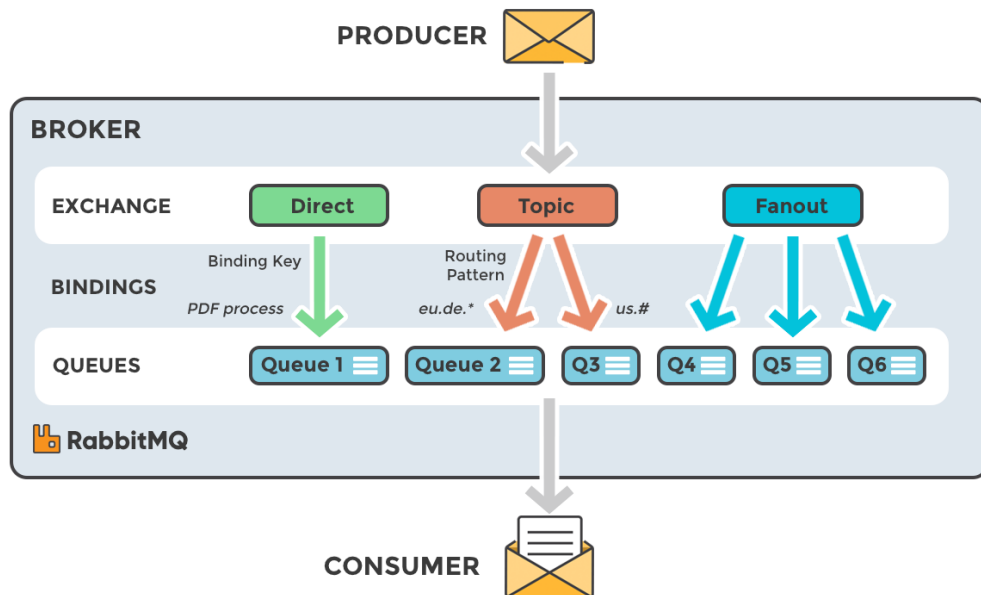


Рисунок 2.5 – Різні способи підписки в RabbitMQ

- посередництво – брокери повідомлень виступають як посередники між відправниками та отримувачами повідомлень; вони приймають повідомлення від відправника та пересилають їх підписникам, не розкриваючи ідентичності або фізичного розташування сторін;
- управління чергами – тимчасове збереження повідомлень у чергах (рисунок 2.6), що дозволяє регулювати темп передачі та надсилання, щоб врахувати навантаження та доступність підписників;

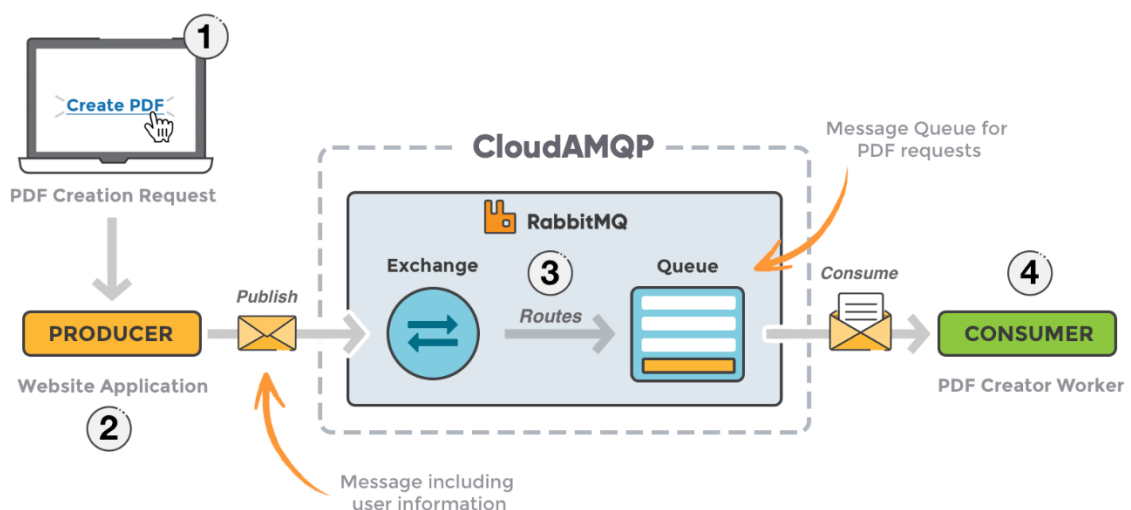


Рисунок 2.6 – Використання черг в RabbitMQ

- гарантія доставки повідомлень – забезпечення гарантії доставки повідомлень, що означає, що повідомлення будуть доставлені до підписників

(рисунок 2.7), навіть якщо деякі сервіси чи компоненти системи тимчасово недоступні.

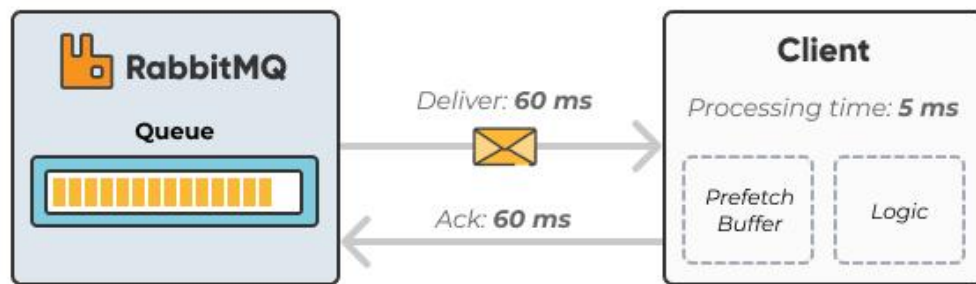


Рисунок 2.7 – Підтвердження (ack) отриманих повідомлень в RabbitMQ

Цей підхід забезпечує надійність та стійкість комунікації в системі.

2.9 Системи асинхронного збереження даних

Системи асинхронного збереження даних використовуються для зберігання та обробки великих обсягів даних з низькою затримкою та високою масштабованістю [25]. Основна ідея полягає у тому, що дані передаються до системи збереження асинхронно, тобто без очікування підтвердження про їх збереження, що дозволяє збільшити швидкість запису та зменшити витрати на операції вводу-виводу.

Переваги асинхронного збереження даних:

- асинхронний запис – дані передаються до системи збереження без очікування підтвердження про їх збереження; дозволяє швидко записувати дані без блокування процесу, що їх генерує;

- буферизація та накопичення – дані можуть бути спочатку накопичені в буфері або журналі, а потім частинами записані на диск або інше сховище; дозволяє оптимізувати операції запису та покращити продуктивність;

- реплікація та резервування – можливість підтримувати реплікацію, тобто створення копій даних на різних вузлах або серверах, що забезпечує високу доступність та стійкість до відмов; крім того, можуть бути використані механізми резервного копіювання для забезпечення безпеки даних;

– масштабованість – можуть бути розподілені на багато вузлів або серверів, що дозволяє обробляти великі обсяги даних та виконувати запити паралельно; дозволяє розподіляти навантаження та підвищує продуктивність системи при обробці великої кількості запитів;

– інтеграція з іншими системами – підтримка інтеграцій з іншими системами та сервісами; дозволяє обмінюватись даними з іншими додатками та використовувати різноманітні інструменти для обробки та аналізу даних;

– відновлення та надійність – можливість відновлення даних в разі відмови або випадку втрати даних; може включати забезпечення резервних копій, механізми реплікації або розподілені сховища для забезпечення надійності та стійкості даних;

– моніторинг та керування дозволяють відстежувати стан системи, виявляти проблеми та здійснювати необхідні налаштування для оптимальної продуктивності та надійності.

Використання систем асинхронного збереження даних дозволяє ефективно зберігати та обробляти великі обсяги даних з низькою затримкою та високою масштабованістю (рисунок 2.8).

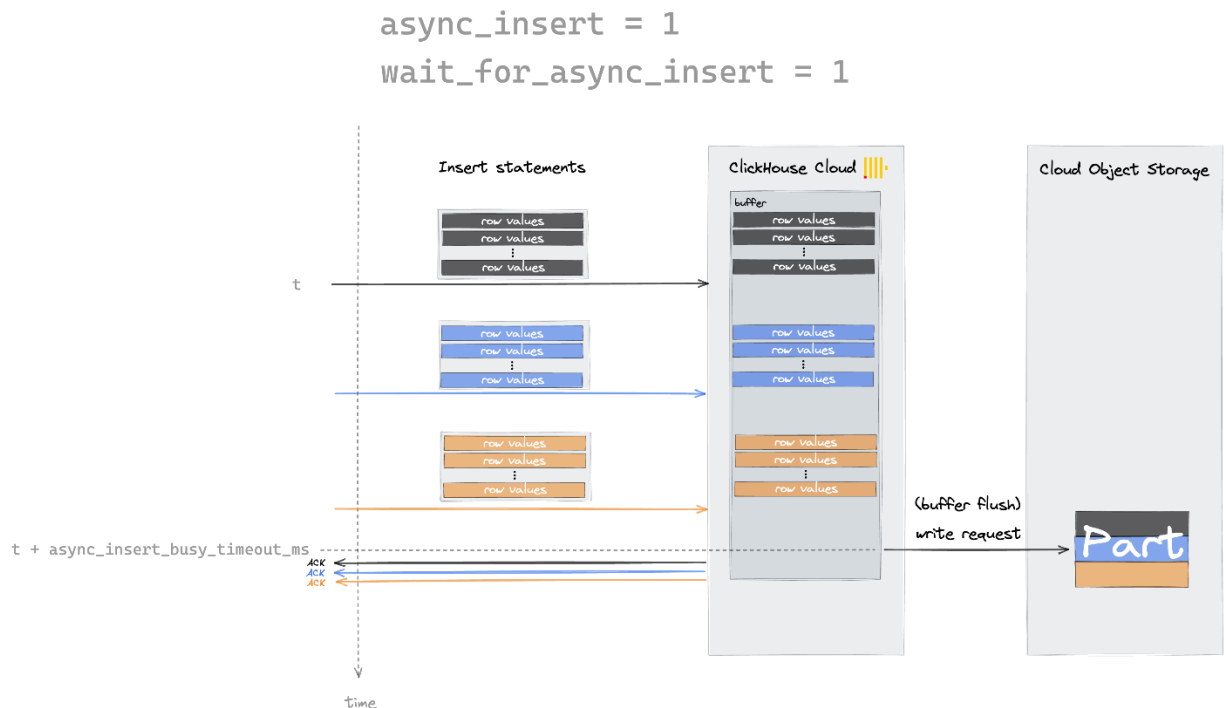


Рисунок 2.8 – Асинхронний запис до ClickHouse

Системи асинхронного збереження забезпечують надійність, відновлюваність та можливість паралельної обробки, що є важливими аспектами розробки систем збору та збереження даних з криптобірж або інших великих об'ємів даних.

2.10 Опис бізнес – процесів

2.10.1 Опис процесу діяльності

Для роботи з криптобіржею Binance скористаємось REST API, через який можна отримувати інформацію про ціни, відкриті та закриті ордери, історію торгів, баланс рахунків та багато іншого. Щоб отримати інформації про останні ціни потрібно використовувати метод, який дає інформацію про останні тикери.

Для отримання інформації про актуальні ціни на криптобіржах зазвичай використовують протоколи зв'язку, які передають дані в режимі реального часу, такі як WebSockets.

Щоб використовувати WebSockets необхідно встановити підключення до API біржі і підписатися на потрібні канали, в яких будуть надходити оновлення цін та іншої інформації. Кожна біржа має свій власний формат даних, який необхідно привести в однаковий формат для коректного використання.

Для використання WebSockets зазвичай потрібно написати код на мові програмування, що підтримує зв'язок по цьому протоколу, і забезпечити безперервну роботу програми, відстежуючи відключення та відновлення з'єднання з API біржі.

За допомогою WebSockets можна отримувати інформацію про курси валют, замовлення, статуси транзакцій та інше (рисунок 2.9). Використання WebSockets допомагає покращити швидкість отримання даних і підвищити точність при торгівлі на криптобіржах.

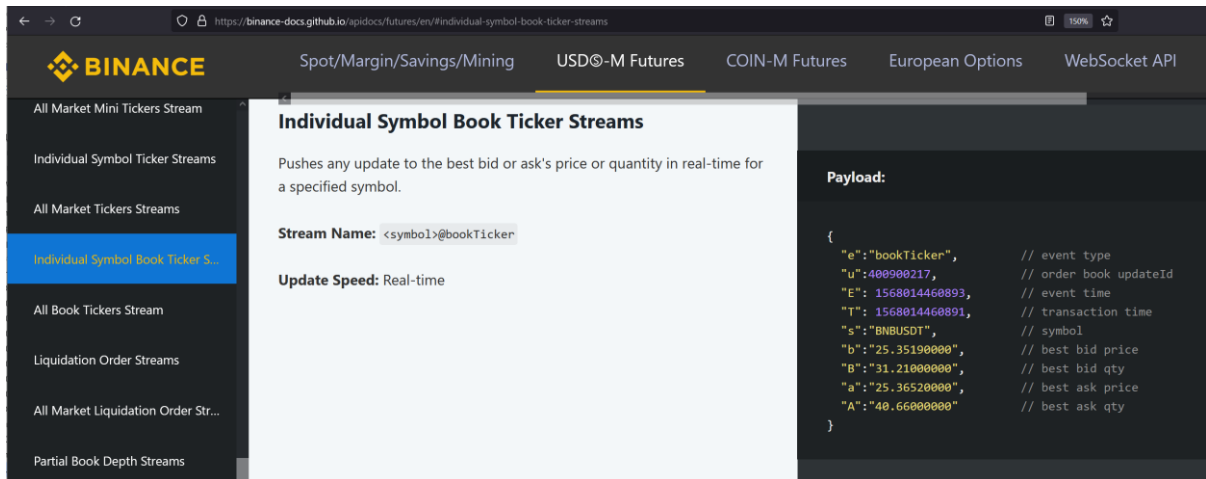


Рисунок 2.9 – Документація Binance WebSockets підключення, для отримання перших цін на купівлю та продаж

Система, яка підписується на тикери через WebSockets, а потім обчислює спред між ними може мати наступний алгоритм дій:

- створити з'єднання з API біржі за допомогою WebSockets;
- підписатись на потік тикерів для потрібних пар криптовалют;
- коли отримуємо новий тикер з кожної пари, зберегти ціни купівлі та продажу для кожної пари;
- обчислити спред між цінами купівлі та продажу для кожної пари;
- застосувати для спреда потрібні обмеження, якщо такі є; наприклад, можна відкинути спреди, які менше певної величини або більше заданого ліміту.

Цей алгоритм можна реалізувати за допомогою сучасною мови програмування, яка підтримує WebSockets та API певної криптобіржі. Крім того, можна додати додаткові функції, такі як управління ризиками, зберігання даних та звітів про прибуток.

Після обчислення спреду між тикерами, дані можна надіслати до черги в RabbitMQ. Для цього потрібно встановити з'єднання з RabbitMQ сервером, створити чергу та відправити повідомлення з отриманими даними в чергу. Це можна зробити за допомогою бібліотеки RabbitMQ для вибраної мови програмування.

Варто враховувати, що в разі надіслання великої кількості повідомлень до черги, може виникнути проблема з їх обробкою. Для запобігання перевантаження сервера, можна використовувати механізм пакетної обробки

повідомлень та контролювати кількість повідомлень, які надсилаються до черги за певний проміжок часу.

Для роботи з RabbitMQ та сховищем даних ClickHouse ми можемо використовувати готові SDK, які надаються розробниками цих систем. Для зберігання даних про спред ми можемо створити таблицю в з необхідними полями (наприклад, назва пари, час, спред відносно попереднього та середнього значення), і надсилати дані до неї через SDK.

Оскільки SDK для роботи з RabbitMQ та обраним сховищем ClickHouse доступні для багатьох мов програмування, використано SDK на мові С.

Основний алгоритм роботи програми буде наступним:

- підписуємось на чергу RabbitMQ та очікуємо повідомлень;
- коли приходить повідомлення, оброблюємо його та отримуємо дані про спред між парами;
- надсилаємо дані до ClickHouse через SDK.

Оскільки дані про спред можуть змінюватись досить часто, можна використовувати батч-вставки для оптимізації запису даних. Також, потрібно налаштувати програму таким чином, щоб вона автоматично приєднувалась до RabbitMQ та сховища даних у разі втрати з'єднання, щоб забезпечити надійну та безперебійну роботу системи.

Графіки спредів можна створити в інструменті візуалізації даних, такому як Grafana, який може підключатись до баз даних, включаючи ClickHouse. Графіки можуть бути налаштовані, щоб відображати спреди в реальному часі або за певний період часу. Графіки можуть бути додані до інтерфейсу користувача для спостереження за рухом спредів та аналізу динаміки ринку. Також можна налаштувати оповіщення, щоб отримувати сповіщення про зміну спредів за певними параметрами.

2.10.2 Актори і функції

Програмний продукт повинен бути повністю автоматизований, таким чином користувач/експерт матиме можливість перегляду лише графіків цін та спредів. В таблиці 2.1 наведено акторів з варіантами використання та описом їх дій.

Таблиця 2.1 – Актори, варіанти використання та їх описи дії

Актор	Варіант використання	Опис дії варіанта використання
Користувач (експерт)	Перегляд графіків цін та спредів	Користувач повинен мати змогу обрати таблицю збереження даних, пару для порівняння та період часу для відображення.
Адміністратор	Підключення нових комбінацій брокерів та підтримка стабільності системи	Адміністратор повинен мати змогу переглядати стан системи у будь- який час, у разі проблеми перезавантажити її та на запит користувача додати інші можливі комбінації брокерів

2.10.3 Структура бізнес-процесів

Структуру бізнес-процесів наведено в додатку А.

2.11 Постановка задачі дослідження

Призначенням розробки даної технології є отримання даних з криптобірж, аналіз їх, розрахунок спреду, покращення його для подальшого використання експертом в торгівельних системах.

Так як система, що реалізує запропоновану технологію працює з відкритим API Binance, то якість зв'язку в деякі моменти часу можуть бути нестабільні, а отже усі данні повинні перевірятись ще на початку отримання з WebSockets з'єднання.

Щоб отримувати данні з різних криптобірж система складається з компонентів. Кожен з яких повинен бути незалежним, щоб в будь який момент можна додати комбінації для аналізу та збільшити спроможність для запису до сховища.

Також для покращення даних потрібно використати інструмент прогнозування/згладжування, так як дані спреду неефективні в чистому вигляді,

бо мають багато «шумів» та різких всплесків та падінь, які заважають експерту чи аналітичній системі приймати рішення.

Тобто ціль розробки: розробити технологію та відповідну систему, яка буде працювати 24/7 без зупинок, з можливістю розширення. Якість даних повинна перевірятись на кожному етапі, та, при збільшенні їх кількості, щоб була можливість перерозподілити навантаження на інші сервери та процеси. І як результат – можливість отримання даних у будь-який момент для прийняття рішення.

Для вирішення поставленої задачі потрібно вирішити наступні питання:

- розробити компонент отримання даних к криптобірж Vinance в режимі реального часу без надлишкового перевантаження API, щоб не бути заблокованим;
- розробити підхід обрахунку спреду та спосіб його аналізу в зручному форматі;
- розробити систему обробки «сирих» даних спреду для подальшого його покращення методами прогнозування та згладжування;
- розробити комплексну архітектуру, яка буде пов'язувати усі компоненти у вигляді динамічної розширюваної системи;
- перевірити навантаження системи при різній кількості працюючих модулів;
- розробити тести для перевірки системи в критичних ситуаціях: зупинка біржі, блокування запитів з її сторони, відмова компонентів системи розташованих на інших серверах;
- перевірити результати отримані експертом в працюючій системі на криптобіржі.

2.12 Розробка компонентів системи

2.12.1 Компонент збору даних та обчислення спреду

Робота сервісу, який збирає дані з криптобіржі, обчислює спред та надсилає їх до RabbitMQ, повинна бути наступною:

- підключення до WebSockets – сервіс встановлює з'єднання з API криптобіржі, яке надає доступ до WebSockets; це дозволяє отримувати потік даних в режимі реального часу про торгові пари, ціни купівлі та продажу, обсяги торгівлі тощо;

- збір та обробка даних – коли з'єднання з WebSockets встановлено, сервіс починає отримувати потік даних; він отримує інформацію про різні торгові пари та їх актуальні ціни з біржі;

- обчислення спреда – після отримання даних, сервіс аналізує ціни купівлі та продажу для кожної торгової пари і обчислює спред;

- відправка даних до RabbitMQ – після обчислення спреда, сервіс використовує клієнт RabbitMQ для надсилання цих даних до RabbitMQ брокера; сервіс може публікувати повідомлення з обчисленими спредами у чергу, яку інші компоненти або сервіси можуть споживати для подальшої обробки або використання.

Особливості сервісу:

- повторення процесу – робота сервісу повинна бути постійною, оскільки ціни на криптобіржі постійно змінюються; тому сервіс повинен підтримувати постійне з'єднання з WebSockets та продовжувати отримувати потік даних про ціни, незважаючи ні на що;

- моніторинг та обробка помилок – сервіс повинен мати механізми моніторингу та обробки помилок, оскільки під час роботи можуть виникати непередбачувані ситуації; наприклад, втрата з'єднання з WebSockets, некоректні дані або проблеми зі з'єднанням з RabbitMQ; у таких випадках сервіс може здійснити повторну спробу встановлення з'єднання або обробити помилку згідно визначених правил;

- масштабованість та оптимізація – залежно від обсягу даних та навантаження, сервіс може бути масштабованим горизонтально або вертикально для оптимальної продуктивності; наприклад, можна використовувати балансувальники навантаження або брокери повідомлень для розподілу роботи між кількома екземплярами;

– безпека та захист даних – сервіс повинен забезпечувати безпеку та захист даних; може включати шифрування комунікації з криптобіржею та RabbitMQ, аутентифікацію та авторизацію для доступу до API біржі та контроль доступу до RabbitMQ для запобігання несанкціонованому доступу до даних.

2.12.2 Компонент запису даних

Робота сервісу, який отримує спред з RabbitMQ та асинхронно зберігає його у ClickHouse, повинна бути наступною:

– підключення до RabbitMQ – сервіс встановлює з'єднання з RabbitMQ брокером, який містить чергу з повідомленнями, що містять дані про спреди; використовуючи відповідний клієнт RabbitMQ, сервіс підписується на чергу, щоб отримувати нові повідомлення;

– отримання та розшифрування повідомлень – коли з'єднання з RabbitMQ встановлено, сервіс починає отримувати повідомлення з черги; кожне повідомлення містить дані про спред, які можуть бути зашифровані або закодовані; сервіс розшифровує або розкодує повідомлення, щоб отримати потрібні дані;

– з'єднання з ClickHouse – після підключення до RabbitMQ, сервіс встановлює з'єднання з базою даних ClickHouse;

– збереження у ClickHouse – після отримання порції даних сервіс асинхронно зберігає дані про спреди у відповідні таблиці бази даних; може включати створення нових записів або оновлення існуючих, залежно від вимог та структури даних.

Особливості сервісу:

– обробка помилок та логування – сервіс повинен мати механізми обробки помилок та відстеження подій; в разі помилки під час збереження даних у ClickHouse або інших проблем, сервіс може повторно спробувати зберегти дані або здійснити відповідні дії залежно від ситуації; крім того, сервіс може здійснювати логування подій, що дозволяє відстежувати операції, виявляти проблеми та проводити аналіз в разі потреби;

– масштабованість та оптимізація – система повинна бути масштабованою, здатною обробляти великі обсяги даних; можуть використовуватися різні методи

для оптимізації продуктивності та ефективного використання ресурсів, такі як паралельна обробка даних, хешування та використання індексів у ClickHouse;

– моніторинг та звітність – сервіс може включати механізми моніторингу для відстеження його стану, продуктивності та виявлення можливих проблем; також можуть бути налаштовані системи звітності, щоб мати детальну інформацію про оброблені дані, час відповіді та будь-які помилки, які виникли під час роботи сервісу;

– безпека та захист даних – сервіс повинен забезпечувати безпеку та захист даних; може включати шифрування комунікації з RabbitMQ та ClickHouse, аутентифікацію та авторизацію для доступу до систем, а також контроль доступу до бази даних та обмеження прав користувачів;

– супровід та підтримка – для ефективної роботи сервісу можуть бути встановлені механізми супроводу та підтримки; це може включати моніторинг виробничого середовища, відстеження оновлень ClickHouse та RabbitMQ для забезпечення актуалізації сервісу, вирішення технічних проблем, а також підтримку з боку команди розробників.

2.13 Рішення з інформаційного забезпечення

Для збільшення швидкості запису даних та можливості інтегрувати в різні системи, формат даних спрощений до одного лінійного запису (рисунок 2.10). Це дає можливість використовувати один формат для всіх компонентів і в ньому ж передавати їх. Якщо в критичні моменти система втратить цілісність сховища, то нові записи не будуть залежати від старих.

spreads	
broker1Symbol	String
broker2Symbol	String
broker1TradeTime	DateTime64
broker2TradeTime	DateTime64
broker1Time	DateTime64
broker2Time	DateTime64
broker1Ask	Float64
broker2Ask	Float64
broker1Bid	Float64
broker2Bid	Float64
spreadAskAsk	Float64
spreadBidBid	Float64

Рисунок 2.10 – ER-діаграма spread

Опис атрибутів:

- broker1Symbol, broker2Symbol – символи (ідентифікатори) даних на криптобіржах;
- broker1TradeTime, broker2TradeTime – фактичний час ціни на криптобіржах;
- broker1Time, broker2Time – час отримання ціни з криптобірж;
- broker1Ask, broker2Ask – ціна першого ордеру на продаж;
- broker1Bid, broker2Bid – ціна першого ордеру на купівлю;
- spreadAskAsk – згладжений спред між першими ордерами на продаж;
- spreadBidBid – згладжений спред між першими ордерами на купівлю.

Для подальшого використання експертом чи аналітичним інструментом, алгоритм обчислення спреду потребує модифікацію для згладження. Цю задачу розглянемо в наступному розділі.

Висновки до розділу

Розглянуто способи збору даних з криптобірж для роботи з різними джерелами та протоколами для отримання потрібної інформації.

Виконаний огляд видів даних з криптобірж, таких як формат даних тикерів, ордербуку та трейдів. Це дозволяє зрозуміти різноманітність даних, які необхідно обробляти та аналізувати.

Висвітлені способи покращення обробки даних з криптобірж, зокрема застосування паралельної потокової обробки, агрегації та перетворення даних, використання брокерів повідомлень та систем асинхронного збереження даних. Це допомагає оптимізувати та прискорити обробку великих обсягів даних з криптобірж.

Особливу увагу націлено на надійності збору даних, а також використанню брокерів повідомлень та систем асинхронного збереження даних для забезпечення стійкості та відновлюваності системи.

Описані бізнес-процеси, їх структура, актори та функції, що дозволяє зрозуміти організацію та взаємодію компонентів системи у контексті роботи з даними з криптобірж.

Розглянуто архітектуру проекту, технології та компоненти, які повинні бути використані. За основу керування навантаження на компоненти обрано RabbitMQ. Одна частина застосунка надсилає до нього результати обчислень, а інша в залежності від кількості даних обмеженими порціями вичитує та формує записи у сховище.

Доступ до сховища обрано асинхронний, тобто в будь-який час інструмент для моніторингу може зробити запит на отримання даних за певний період, і ця дія не заблокує записи нових даних.

Для отримання даних є можливість запуску незалежних сервісів, які потім налаштовуються в одну мережу і як наслідок – додаткова верифікація отриманої інформації.

Розроблено структуру даних для використання в усіх компонентах системи.

3 ПІДХОДИ ДО ЗГЛАДЖУВАННЯ СПРЕДУ ЦІНИ

3.1 Постановка задачі згладжування спреду цін

Основна ідея застосування згладжування спреду ціни [26] полягає у тому, щоб зменшити шум і непередбачуваність в цінах, що можуть виникнути через велику кількість різних факторів, які можуть виникнути в нестабільному середовищі крипторинку.

Основні причини використання методів згладження для нашої системи:

- покращення прогнозування [28] – так як зменшується вплив короткострокової випадкової змінної на ціну;
- підвищення точності – завдяки зменшенню коливання ціни, пов'язані зі змінами в економічному середовищі;
- зменшення шуму – зменшення шуму, пов'язаного з непередбачуваними змінами в зовнішньому середовищі, і як результат – управління ризиками стає більш контрольованим;
- зниження ризику – бо крипторинки має високу волатильність [27], оскільки воно зменшує коливання ціни і зробить вплив ринку менш непередбачуваним.

3.2 Аналіз методів згладжування

3.2.1 *Simple Exponential Smoothing*

Просте експоненційне згладжування [29] – метод, що використовується для прогнозування часових рядів, що мають експоненційно зменшувану залежність.

Принцип методу полягає в тому, що наступне значення в часовому ряді прогнозується шляхом зваженого середнього минулих значень ряду, де вага найбільшого значення відводиться останньому спостереженню, а ваги зменшуються експоненційно зі збільшенням віку спостережень. Основні параметри методу SES - це коефіцієнт згладжування α ($0 \leq \alpha \leq 1$), що визначає, наскільки сильно поточне прогнозоване значення залежить від

попередніх значень, та початкове значення ряду (перше спостереження), яке може бути задане як випадкова величина або як середнє значення ряду.

Цей метод використовується для того, щоб зменшити вплив шуму і випадкових змін на прогноз. Обчислення відбуваються у такий спосіб:

$$\hat{y}_{t+h|t} = l_t,$$

$$l_t = \alpha y_t + (1-\alpha)l_{t-1},$$

де h – крок прогнозування, l_t – згладжене(прогнозоване) значення в час t , α – параметр згладжування.

Крім того цей метод дуже швидкий у розрахунках та не потребує значного обсягу обчислювальних ресурсів.

3.2.2 Holt's Linear Method

Метод Хольта [31] – це розширений метод експоненційного згладжування, який включає в себе дві компоненти – рівень та тренду.

Цей метод моделює часовий ряд як комбінацію рівнів та трендів компонентів і використовує два параметри згладжування: α (параметр згладжування рівнів) та β^* (параметр згладжування тренду), для оцінки цих компонентів. Метод припускає, що рівень та тренду часового ряду змінюються лінійно з часом.

Метод включає такі кроки:

- ініціалізація – початкові оцінки рівня та тренду розраховуються на основі перших двох спостережень в часовому ряді;
- згладжування рівня та тренду – оцінки рівня та тренду оновлюються для кожного часового періоду на основі попередніх оцінок та поточного спостереження;
- прогнозування – майбутні значення прогнозуються за допомогою оцінених компонентів рівня та тренду;
- оновлення параметрів згладжування – параметри згладжування коригуються на основі помилки між фактичними та прогнозованими значеннями.

Однією з переваг методу Хольта є те, що він може обробляти дані з тренду, на відміну від простого експоненційного згладжування [30]. Метод Хольта

доцільно використовувати для прогнозування часових рядів, що мають лінійний тренд.

Обчислення відбуваються за такими виразами:

$$\begin{aligned}\hat{y}_{t+h|t} &= l_t + hb_t, \\ l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}), \\ b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1},\end{aligned}$$

де b_t – значення тренду у час t , β^* – параметр згладжування тренду.

3.2.3 Additive Damped Trend Method

Адитивне згладжування тренду зі згасанням [33] – це метод, що використовується для прогнозування часових рядів, які мають тренд та сезонні зміни.

Метод передбачає, що тренд буде згасати до деякого фіксованого значення в майбутньому. Це означає, що нові спостереження будуть мати менший вплив на тренд, ніж старіші. Така модель може бути корисною для прогнозування часових рядів з трендом, який може змінюватися у часі, але зміна його не є довгостроковою.

Для прогнозування метод використовує оцінені рівень та тренд, а також оцінку згасаючого коефіцієнту тренду. Цей коефіцієнт використовується для того, щоб врахувати зміну тренду в часі. Крім того, метод дозволяє розраховувати інтервали довіри для прогнозованих значень.

Однією з переваг методу додаткового згладжування зі згасаючою тенденцією є його здатність враховувати зміну тренду в часі та зменшувати її вплив з часом. Крім того, метод може бути корисним для прогнозування часових рядів, які мають тренд, що змінюється у часі.

Розрахунок здійснюється за рівняннями:

$$\begin{aligned}\hat{y}_{t+h|t} &= l_t + \phi b_t, \\ l_t &= \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1}), \\ b_t &= \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}.\end{aligned}$$

де ϕ – параметр згладжування.

3.3 Результати досліджень ефективності методів

Для порівняння досліджуваних методів був обраний період з 10 березня 2023 року (timestamp – 1678449657000) до 15 березня 2023 року (1678881657000).

Вхідні значення цін зображено на рисунку 3.1, а різниця між ними (спред) – на рисунку 3.2.

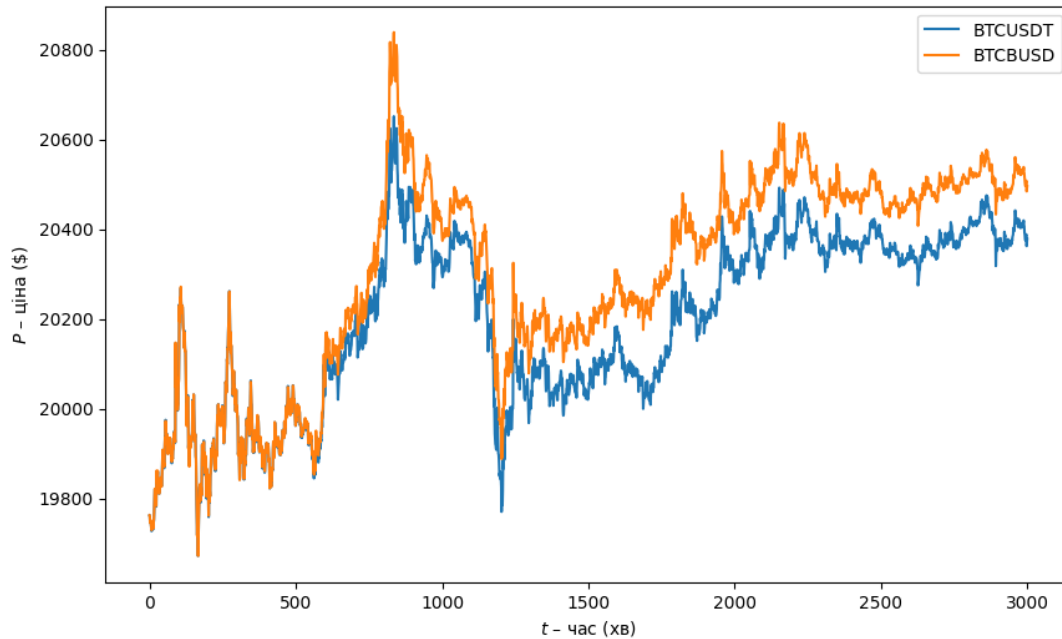


Рисунок 3.1 – Зміна ціни криптовалютних пар BTCUSDT та BTCBUSD на біржі Binance.

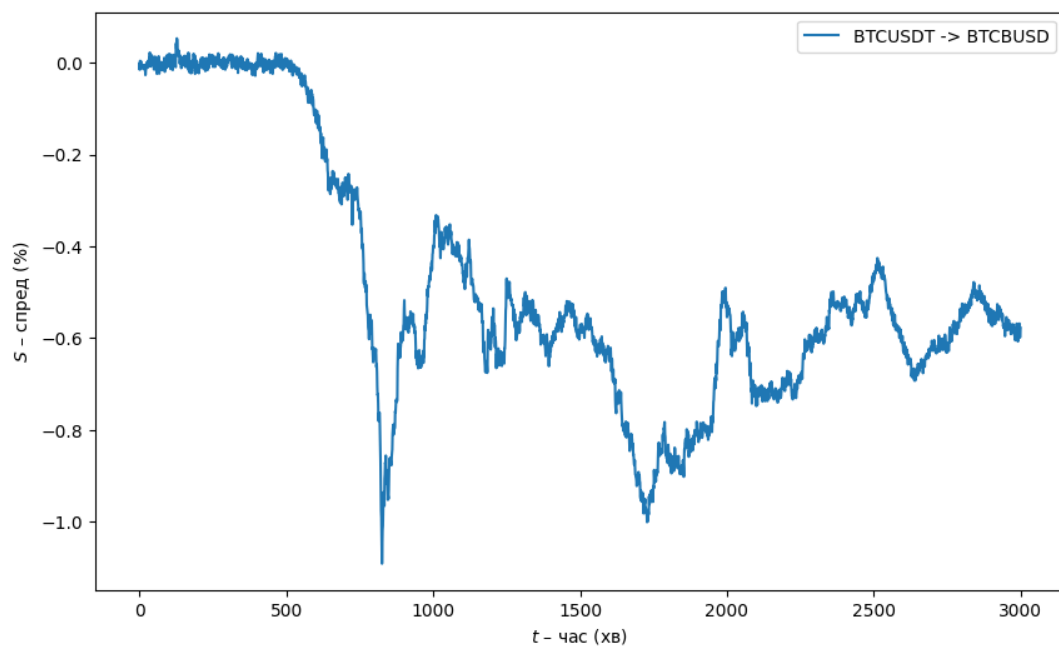


Рисунок 3.2 – Зміна спреда між криптовалютними парами BTCUSDT та BTCBUSD на біржі Binance.

Результати часу роботи досліджуваних алгоритмів наведені у таблицях 3.1, 3.2, 3.3.

Таблиця 3.1 – Час згладжування Simple Exponential Smoothing з різними вхідними параметрами.

Розмір блоку	Рівень згладжування	Час виконання (мс.)
100	0.100	5089.386940
300	0.100	5711.791039
500	0.100	6430.731773
100	0.010	5050.818920
300	0.010	5766.963959
500	0.010	6504.789829
100	0.001	5109.390020
300	0.001	5798.051119
500	0.001	6436.933756

Таблиця 3.2 – Час згладжування Holt's Linear Method з різними вхідними параметрами.

Розмір блоку	Рівень згладжування	Час виконання (мс.)
100	0.100	8259.335041
300	0.100	14654.006004
500	0.100	20657.027960
100	0.010	8378.038883
300	0.010	14696.868896
500	0.010	20736.852884
100	0.001	8385.403156
300	0.001	14770.668983
500	0.001	20638.543129

Таблиця 3.3 – Час згладжування Additive Damped Trend Method з різними вхідними параметрами.

Розмір блоку	Рівень згладжування	Час виконання (мс.)
100	0.100	14226.014853
300	0.100	20629.573107
500	0.100	26756.132841
100	0.010	15543.703079
300	0.010	21791.299105
500	0.010	27713.525057
100	0.001	15884.458780
300	0.001	23743.767023
500	0.001	29761.780977

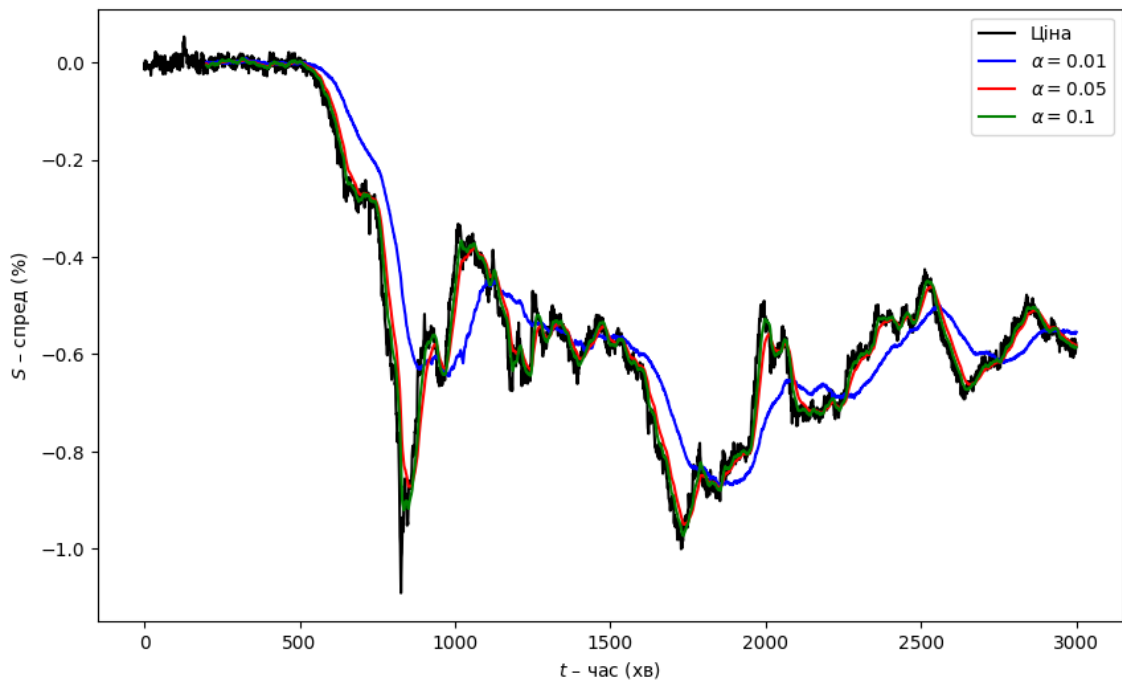


Рисунок 3.3 – Зладжений спред методом Simple Exponential Smoothing з розміром блоку в 200.

За результатом таблиць 3.1, 3.2, 3.3 можна зробити висновок, що просте експоненційне згладжування має значну перевагу за швидкість серед усіх переглянутих методів.

Висновки до розділу

Досліджені три методи експоненційного згладжування, в яких не має сезонної залежності: Simple Exponential Smoothing, Holt's Linear Method, Additive Damped Trend. Так як для аналітичного відображення коефіцієнт α повинен бути менше 0.01, то отриманий результат майже однаковий. Таким чином за критерій оцінки обрано час обчислення, де перший простий метод показав найкращий результат.

За результатом дослідження для модифікації формули обрахунку спреду доцільно використовувати Simple Exponential Smoothing.

4 РОЗРОБКА ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до програмного продукту

Програмний продукт для експерту повинен бути представлений як графічний термінал у браузері за допомогою інструменту Grafana.

При завантаженні та обробці даних з криптобірж важливо дотримуватись певних вимог, щоб забезпечити якісну та надійну обробку інформації:

- повнота та актуальність – дані повинні бути завантажені повністю і не повинні містити пропусків або втрати інформації; крім того, дані мають бути актуальними, оновлюватися в режимі для забезпечення актуальності для результуючого аналізу;

- якість даних – дані повинні бути якісними, маючи високу точність та достовірність; помилки та неточності можуть спотворити результати аналізу та призвести до невірних висновків;

- швидкодія – завантаження та обробка даних повинні здійснюватися ефективно та вчасно; використання порівняно швидких алгоритмів може допомогти підвищити швидкодію обробки даних;

- масштабованість – залежно від обсягу даних і потреб аналізу, система повинна мати можливість масштабуватися, щоб впоратись зі збільшенням обсягів даних;

- інтеграція з іншими системами – система обробки даних повинна бути здатна інтегруватися з іншими системами або інструментами, які використовуються для аналізу даних; застосування інтеграції з платформами візуалізації даних або інструментами для проведення аналізу та отримання прогнозів.

4.2 Засоби розробки

4.2.1 VS Code

Visual Studio Code (VS Code) – це популярний текстовий редактор файлів, розроблений компанією Microsoft, який використовують для розробки ПЗ. Це

безкоштовне та відкрите програмне забезпечення, яке працює на Windows, macOS та Linux.

VS Code має сучасний та зрозумілий інтерфейс користувача (рисунок 4.1), який можна налаштувати під свої потреби. Він підтримує широкий спектр мов програмування та має вбудовані функції, такі як підсвічування синтаксису, автодоповнення коду, налагодження та інтеграцію з системами контролю версій.

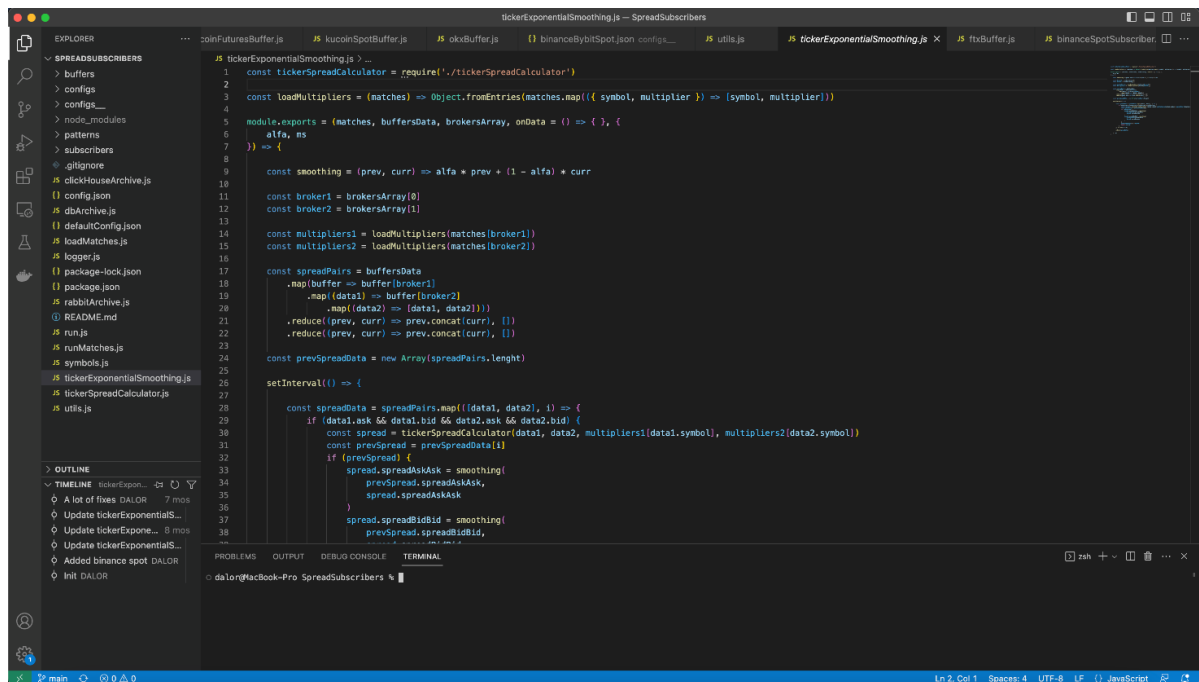


Рисунок 4.1 – Проект сервісу завантаження та обчислення спреду

Інша важлива функція VS Code – це термінал інтегрований в IDE, який дозволяє розробникам виконувати команди та скрипти, не покидати редактор.

4.2.2 Node.JS

Node.js – це середовище виконання JavaScript на стороні сервера, яке має багато переваг. Node.js побудований на движку V8, що забезпечує швидкість виконання коду, а однопоточна, асинхронна модель виконання дозволяє ефективно використовувати ресурси сервера.

Додатки, побудовані на Node.js, можна легко масштабувати вертикально та горизонтально, збільшуючи кількість серверів або їх потужність. Node.js є відкритим джерелом, тому є безкоштовним та доступним для всіх, і має велику спільноту розробників, яка активно підтримує його та надає допомогу. Однією з найбільших переваг Node.js є наявність великої кількості модулів та бібліотек, які дозволяють розробникам швидко та ефективно вирішувати різноманітні

задачі. Багата інфраструктура та велика спільнота Node.js забезпечують наявність багатьох інструментів та розширень, що дозволяють розробникам легко протестувати, налагодити та оптимізувати свої додатки.

4.2.3 WebSockets

WebSockets – це протокол, який дозволяє встановити постійне двостороннє з'єднання між браузером та веб-сервером. Він дозволяє обмінюватися даними між клієнтом та сервером в режимі реального часу. Звичайний HTTP-протокол не може забезпечити такої функціональності, оскільки він працює на основі запит-відповідь, тобто клієнт відправляє запит на сервер, а сервер відповідає на запит. Порівняння в способі передачі даних наведено на рисунку 4.2.

Він дозволяє розробникам створювати більш інтерактивні та динамічні веб-додатки, які можуть швидко та ефективно обмінюватися даними з сервером та іншими клієнтами. WebSockets є ефективним рішенням для завдань, які вимагають швидкої передачі даних та реагування на дії користувача.

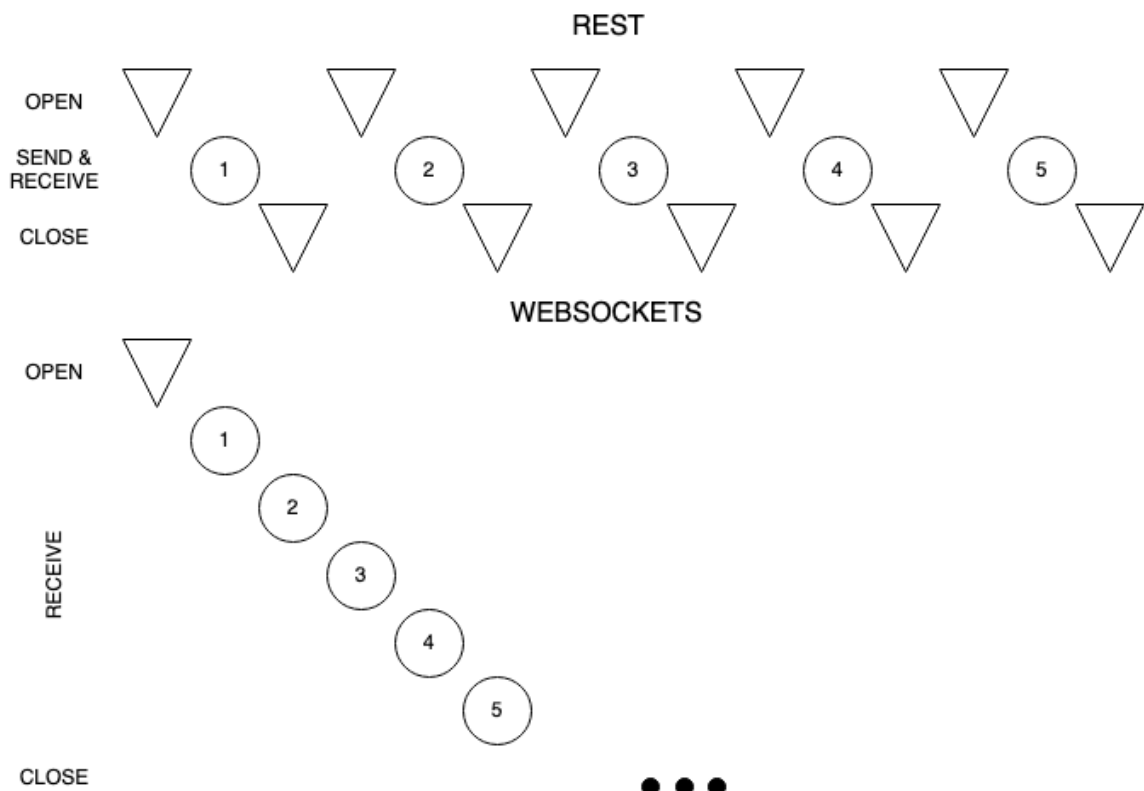


Рисунок 4.2 – Перевага в швидкості отримання даних через WebSockets над REST

4.2.4 RabbitMQ

RabbitMQ – це програмне забезпечення, яке забезпечує передачу повідомлень між різними компонентами програмної системи. Ця технологія використовує протокол AMQP (Advanced Message Queuing Protocol), що дозволяє забезпечити ефективну та безпечну передачу повідомлень між різними компонентами програмної системи, незалежно від мови програмування та платформи.

RabbitMQ забезпечує можливість створювати різні черги повідомлень (рисунок 4.3), до яких можуть підключатись різні компоненти програмної системи. Кожен компонент може відправляти та отримувати повідомлення з черги, що дозволяє ефективно організувати роботу компонентів та зменшувати навантаження на сервер.

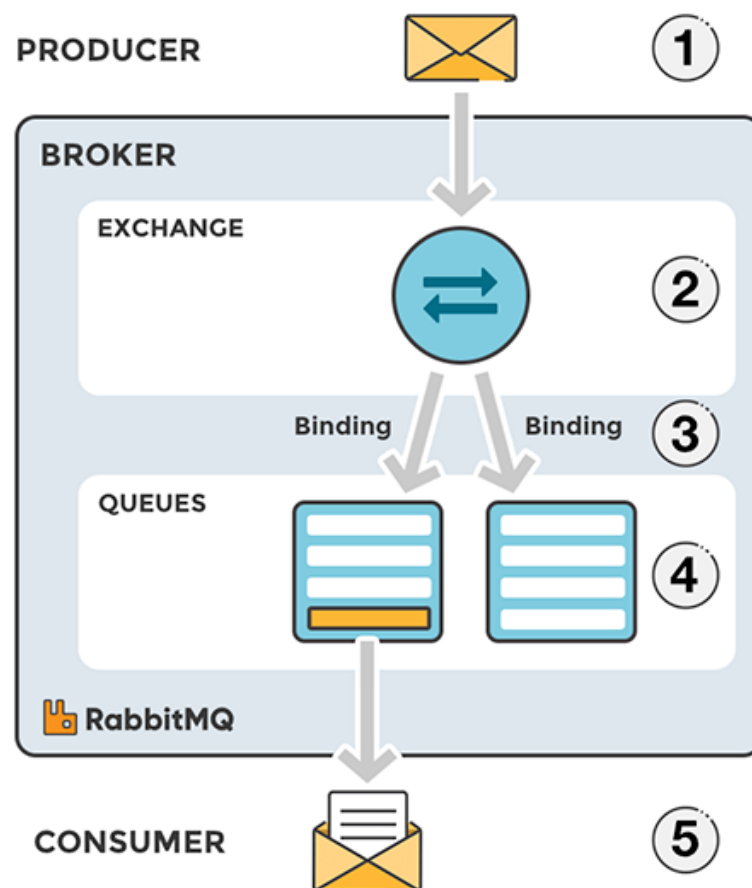


Рисунок 4.3 – Приклад розподілення навантаження в RabbitMQ

RabbitMQ знаходить своє застосування в багатьох галузях, таких як фінанси, медицина, телекомунікації та інші. Вона дозволяє підвищити

ефективність та масштабованість систем, які працюють з великою кількістю повідомлень (рисунок 4.4), та забезпечити надійну доставку повідомлень між різними компонентами програмної системи.

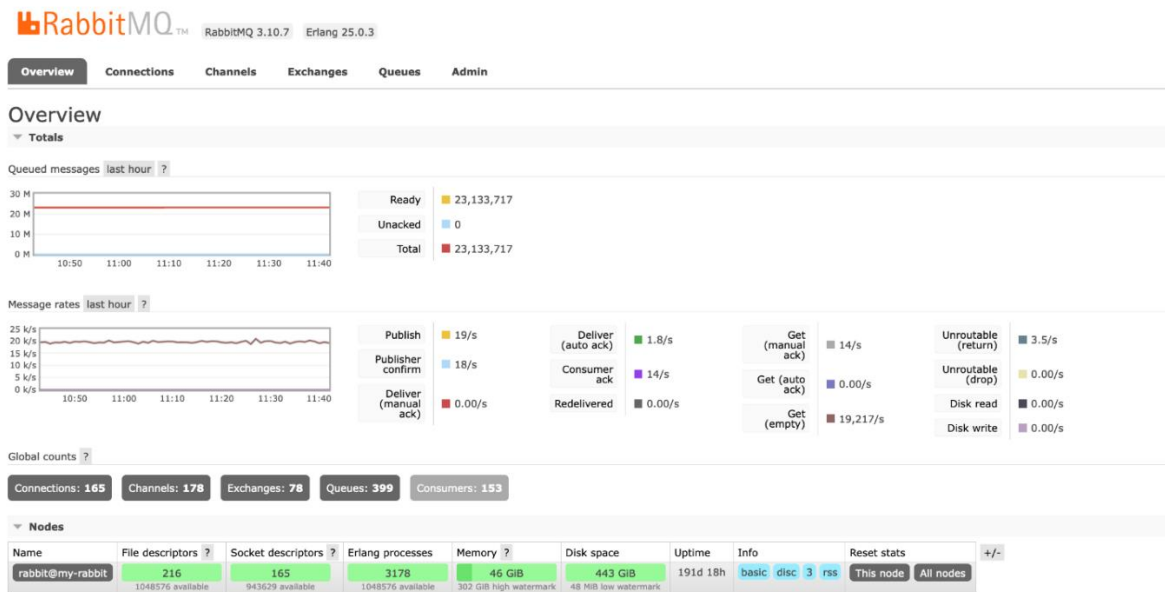


Рисунок 4.4 – Сторінка моніторингу навантаження RabbitMQ

З рисунку 4.4 видно, що RabbitMQ може спокійно працювати з 23 мільйонами повідомлень.

4.2.4 ClickHouse

ClickHouse – це відкрите програмне забезпечення для аналізу та обробки даних, яке спеціально розроблене для роботи з великими обсягами даних в реальному часі. Ця технологія дозволяє зберігати та обробляти мільярди рядків даних за декілька секунд на одному сервері.

ClickHouse використовує колонкову орієнтовану модель зберігання даних (рисунок 4.6), яка забезпечує швидкий доступ до даних та ефективну обробку запитів. Вона використовує векторну обробку даних та спеціальну оптимізацію для масштабованої обробки даних в режимі реального часу. ClickHouse також підтримує різні формати даних, включаючи CSV, JSON.

The screenshot shows the Tabix web interface for ClickHouse. On the left, a tree view shows the database structure under 'storage1', including 'INFORMATION_SCHEMA', 'default', 'information_schema', 'rabbit2ch1', and 'spreads'. The 'spreads' table is selected. The main area displays a SQL query:

```

1 SELECT
2   broker1Symbol,
3   broker2Symbol,
4   broker1TradeTime,
5   broker2TradeTime,
6   broker1Time,
7   broker2Time,
8   broker1Ask,
9   broker2Ask,
10  broker1Bid,
11  broker2Bid,
12  spreadAskAsk,
13  spreadBidBid
14 FROM
15   spreads.binance_binance_futures_futures_spreads
17 LIMIT 100

```

Below the query, a table of results is shown. The table has 13 columns corresponding to the query fields. The first few rows are:

broker1Symbol	broker2Symbol	broker1TradeTime	broker2TradeTime	broker1Time	broker2Time	broker1Ask	broker2Ask	broker1Bid	broker2Bid	spreadAskAsk	spreadBidBid
SOLUSD	SOLBUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	17.927	17.83	17.924	17.829	0.544026920...	0.532839755...
ETHUSD	ETHUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	1312.47	1316.21	1312.46	1316.2	-0.28414918...	-0.28415134...
SOLUSD	SOLUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	17.83	17.927	17.829	17.924	-0.54108328...	-0.53801562...
MATICUSD	MATICUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	1.0917	1.0887	1.0916	1.0886	0.275558004...	0.275583318...
ETHUSD	ETHUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	1316.21	1312.47	1316.2	1312.46	0.284958894...	0.284961865...
MATICUSD	MATICUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	1.0887	1.0917	1.0886	1.0916	-0.27480076...	-0.27482594...
MATICUSD	MATICUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	1.0887	1.0917	1.0886	1.0916	-0.27480076...	-0.27482594...
GALAUUSD	GALABUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	0.03161	0.031558	0.0316	0.031536	0.164775968...	0.164775968...
GALAUUSD	GALABUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	0.03161	0.031558	0.0316	0.031536	0.164775968...	0.164775968...
GALAUUSD	GALABUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	0.03161	0.031557	0.0316	0.031536	0.164781046...	0.164781046...
GALAUUSD	GALABUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	0.03161	0.031557	0.0316	0.031536	0.164781046...	0.164781046...
OKX_FUTURES_SPREADS	SANDBUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	0.6665	0.6684	0.6657	0.6682	-0.28426092...	-0.37413947...
OKX_FUTURES_SPREADS	SANDBUSD	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	2022-11-10 ...	0.6665	0.6684	0.6657	0.6682	-0.28426092...	-0.37413947...

Рисунок 4.5 – Tabix – інструмент для роботи з ClickHouse через Web

The screenshot shows the Tabix web interface displaying the schema of the 'spreads' table. The table name 'spreads' is shown with a count of 36. The columns and their data types are:

Column Name	Data Type
broker1Symbol	String
broker2Symbol	String
broker1TradeTime	DateTime64(3)
broker2TradeTime	DateTime64(3)
broker1Time	DateTime64(3)
broker2Time	DateTime64(3)
broker1Ask	Float64
broker2Ask	Float64
broker1Bid	Float64
broker2Bid	Float64
spreadAskAsk	Float64
spreadBidBid	Float64

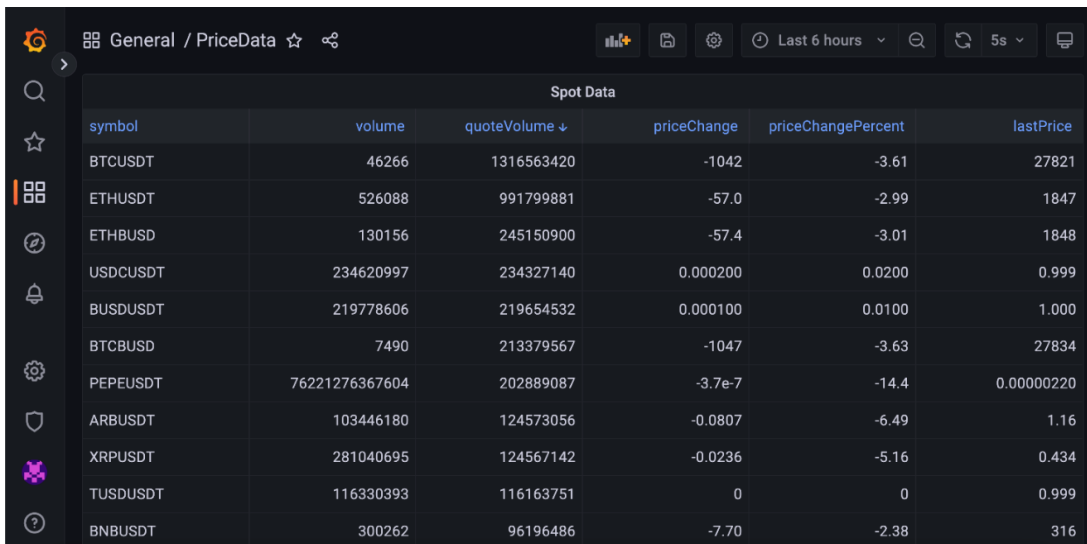
Рисунок 4.6 – Таблиця збереження даних в ClickHouse

ClickHouse дозволяє розробникам створювати швидкі та ефективні системи аналізу даних, які можуть обробляти великі обсяги даних в реальному часі. Завдяки високій продуктивності та масштабованості, ClickHouse допомагає організаціям зменшити витрати на обробку та аналіз даних та підвищити ефективність бізнесу.

4.2.5 Grafana

Grafana – це інструмент візуалізації даних та моніторингу, який дозволяє аналізувати та відстежувати стан системи в реальному часі. Ця технологія надає можливість користувачам створювати різноманітні графіки, таблиці та панелі для відображення даних з різних джерел, таких як бази даних, сервери моніторингу та інші системи.

Grafana має простий та інтуїтивно зрозумілий інтерфейс, який дозволяє швидко створювати та налаштовувати візуалізацію даних (рисунок 4.7). Користувачі можуть використовувати різноманітні типи діаграм, включаючи лінійні (рисунок 4.8), кругові та стовпчикові діаграми, для представлення даних у зручному форматі.



symbol	volume	quoteVolume ↓	priceChange	priceChangePercent	lastPrice
BTCUSDT	46266	1316563420	-1042	-3.61	27821
ETHUSDT	526088	991799881	-57.0	-2.99	1847
ETHBUSD	130156	245150900	-57.4	-3.01	1848
USDCUSDT	234620997	234327140	0.000200	0.0200	0.999
BUSDUSDT	219778606	219654532	0.000100	0.0100	1.000
BTCBUSD	7490	213379567	-1047	-3.63	27834
PEPEUSDT	76221276367604	202889087	-3.7e-7	-14.4	0.00000220
ARBUSDT	103446180	124573056	-0.0807	-6.49	1.16
XRPUSDT	281040695	124567142	-0.0236	-5.16	0.434
TUSDUSDT	116330393	116163751	0	0	0.999
BNBUSDT	300262	96196486	-7.70	-2.38	316

Рисунок 4.7 – Приклад табличного відображення в Grafana



Рисунок 4.8 – Приклад графіку в Grafana

Grafana також дозволяє встановлювати сповіщення на певні події, такі як перевищення порогових значень, що дозволяє оперативно відстежувати стан системи та реагувати на проблеми.

Ця технологія знаходить широке застосування в сфері моніторингу та аналізу даних, включаючи моніторинг стану мережі, системи, серверів, додатків та інших компонентів. Завдяки зручному та ефективному інтерфейсу, Grafana дозволяє користувачам отримувати швидкий та зрозумілий доступ до даних про стан системи та аналізувати їх в режимі реального часу.

4.3 Архітектура програмного забезпечення

Архітектура проекту – мікросервісна. Таким чином кожен компонент має свою роль і є частиною єдиної системи:

- компонент обчислення спреду – цей компонент відповідає за завантаження даних з Binance, обчислення спреду та відправку результатів до RabbitMQ; він надсилає усі результати до черги RabbitMQ;

- компонент запису даних – цей компонент відповідає за отримання повідомлень з RabbitMQ, які містять дані спреду, і записує їх до ClickHouse; він групами отримує повідомлення з черги RabbitMQ;

- ClickHouse – виступає як сховище для зберігання даних про спред; компонент запису даних виконує асинхронний доступ до ClickHouse для запису нових даних; інструмент моніторингу також може виконувати асинхронний запит до ClickHouse для отримання даних за певний період;

- RabbitMQ використовується для забезпечення асинхронного зв'язку між компонентами системи; компонент обчислення спреду надсилає результати обчислень у чергу RabbitMQ, а компонент запису даних споживає ці повідомлення з черги та зберігає їх до ClickHouse;

- графічний інтерфейс Grafana використовується для моніторингу та візуалізації даних; він підключається до ClickHouse для отримання даних та створення графіків, панелей та інших візуалізацій.

4.4 Інструкція користувача

4.4.1 Підготовка середовища для сервісу зберігання спредів. Встановлення `g++` компілятора

Для запуску C/C++ додатків потрібно встановити компілятор `g++`:

– перевірте, чи вже встановлений компілятор `g++` на вашій системі, введіть команду:

```
g++ --version
```

– якщо ви отримали відповідь з версією компілятора, це означає, що `g++` вже встановлений. В іншому випадку продовжуйте наступними кроками;

– встановіть компілятор `g++` на свою систему:

Ubuntu / Debian – виконайте наступну команду в терміналі:

```
sudo apt-get update
```

```
sudo apt-get install g++
```

Fedora / CentOS – виконайте наступну команду в терміналі:

```
sudo dnf install gcc-c++
```

MacOS – для Homebrew виконайте наступну команду в терміналі:

```
brew install gcc
```

4.4.2 Встановлення `rabbitmq-c`

Для встановлення `rabbitmq-c`, бібліотеки для роботи з RabbitMQ на мові C, потрібно:

– завантажити код `rabbitmq-c` з офіційного репозиторію на GitHub; можна зробити це за посиланням: <https://github.com/alanxz/rabbitmq-c>;

– змініть поточну робочу директорію в терміналі на папку завантаженого проекту `rabbitmq-c`;

– у папці проекту знайдіть файл `configure`, із якого ви можете згенерувати файли конфігурації для вашої системи; запустіть наступну команду в терміналі:

```
./configure
```

Ця команда визначить вашу систему та налаштує параметри збірки для RabbitMQ-C;

– після успішного виконання попередньої команди виконайте наступну команду для збирання та встановлення RabbitMQ-C:

```
make && make install
```

Цей крок встановить RabbitMQ-C у вашій системі.

4.4.2 Встановлення *clickhouse-cpp*

Для встановлення ClickHouse Client для C++ потрібно:

– завантажити код ClickHouse Client для C++ з офіційного репозиторію на GitHub. Можна зробити це за посиланням: <https://github.com/ClickHouse/ClickHouse-CPP>;

– змініть поточну робочу директорію в терміналі на папку завантаженого проекту;

– створіть папку `build` у директорії проекту та змініть поточну робочу директорію в терміналі на папку `build`;

– виконайте наступну команду для генерації файлів конфігурації засобу CMake:

```
cmake ..
```

– після успішного виконання попередньої команди виконайте команду збирання проекту:

```
make && make install
```

4.4.3 Компіляція сервісу збереження даних

Перед компіляцією спочатку потрібно налаштувати його. Для цього в файлі конфігурації потрібно встановити параметри, які не будуть змінюватись для вашої системи:

```
– char const *rabbitHost = "localhost";
```

```
– int rabbitPort = 5671;
```

```
– char const *rabbitExchange = "spreads";
```

```
– char const *rabbitUser = "guest";
```

```
– char const *rabbitPassword = "guest";
```

```
– char const *clickhouseHost = "localhost";
```

```
– int clickhousePort = 9000;
```

```
– char const *clickhouseUser = "default";
```

– char const *clickhousePassword = "default";

– char const *clickhouseDatabase = "spreads";

В таблиці 4.1 наведено опис цих параметрів.

Таблиця 4.1 – Опис параметрів запуску сервіса збереження даних

Назва параметру	Тип	Базове значення	Опис
rabbitHost	Текст	localhost	Адреса серверу RabbitMQ
rabbitPort	Число	5671	Порт серверу RabbitMQ
rabbitUser	Текст	guest	Ім'я користувача серверу RabbitMQ
rabbitPassword	Текст	guest	Пароль користувача серверу RabbitMQ
clickhouseHost	Текст	localhost	Адреса серверу ClickHouse
clickhousePort	Число	9000	Порт серверу ClickHouse
clickhouseUser	Текст	default	Ім'я користувача серверу ClickHouse
clickhousePassword	Текст	default	Пароль користувача серверу ClickHouse
clickhouseDatabase	Текст	spreads	Ім'я бази даних в ClickHouse, де будуть створюватись таблиці

Залишимо лише 1 динамічний параметр, який буде відповідати лише за назву черги в RabbitMQ та таблиці в ClickHouse.

Далі виконаємо команду компіляції:

```
g++ main.cpp
```

Для запуску скрипту треба його просто запустити:

```
./main
```

4.4.4 Підготовка середовища для сервісу завантаження цін та обчислення спредів. Встановлення Node.js

Щоб встановити Node.js 16 за допомогою терміналу, слід дотримуватися наступних кроків:

– встановіть Node Version Manager – NVM є інструментом, який дозволяє керувати версіями Node.js на вашому комп'ютері. Він дозволить вам встановити Node.js 16 і легко перемикається між версіями за потребою. Введіть наступну команду в терміналі для встановлення NVM:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

Ця команда завантажить та встановить NVM на ваш комп'ютер;

– після встановлення NVM ви можете встановити Node.js 16. Введіть наступну команду в терміналі:

```
nvm install 16
```

Ця команда завантажить і встановить Node.js 16 на ваш комп'ютер;

– щоб перевірити, чи встановлено Node.js 16, введіть наступну команду:

```
node -v
```

Якщо ви бачите вивід, який починається з "v16.x.x", це означає, що встановлення Node.js 16 – успішне.

4.4.5 Налаштування та запуск сервісу завантаження цін та обчислення спредів.

Для налаштування сервісу потрібно створити файл з розширенням .json та наступним вмістом:

```
{
  "rabbit": {
    "url": "amqp://guest:guest@localhost",
    "exchange": "spreads",
    "routingKey": "binance_binance_futures_futures_spreads",
```

```

    "asColumns": false
  },
  "brokers": [
    "binance",
    "binance"
  ]
}

```

Параметр `brokers` – це список з двох параметрів, які вказують на комбінацію та порядок порівняння даних з прописаних джерел.

Параметр `rabbit` – це конфігурація підключення до RabbitMQ.

В таблиці 4.2 наведено параметри, які потрібні для запуску компоненту обчислення спреду.

Таблиця 4.2 – Опис параметрів запуску сервіса обчислення спреду

Назва параметру	Тип	Базове значення	Опис
<code>url</code>	Текст	<code>amqp://guest:guest@localhost</code>	Адреса серверу RabbitMQ, в якій повинно бути також описано логін користувача, його пароль та порт серверу
<code>exchange</code>	Текст	<code>spreads</code>	Назва провайдеру для RabbitMQ, на який будуть посилатись черги для підписки
<code>routingKey</code>	Текст	<code>binance_binance_futures_futures_spreads</code>	Шлях підписки на провайдер для RabbitMQ, на який будуть надсилатись дані обраного типу

asColumns	Булевий	false	false – надсилати дані построково; true – у вигляді транспонованої матриці
-----------	---------	-------	--

Перед запуском потрібно оновити усі бібліотеки. Для цього в каталозі проекту потрібно виконати команду:

```
npm install
```

Для запуску:

```
node run.js НАЗВА_КОНФІГУРАЦІЙНОГО_ФАЙЛУ.json
```

4.4.5 Налаштування Grafana для відображення спредів.

Спочатку створіть нову дошку (рисунок 4.9), де буде відображатись інформація про спреди.

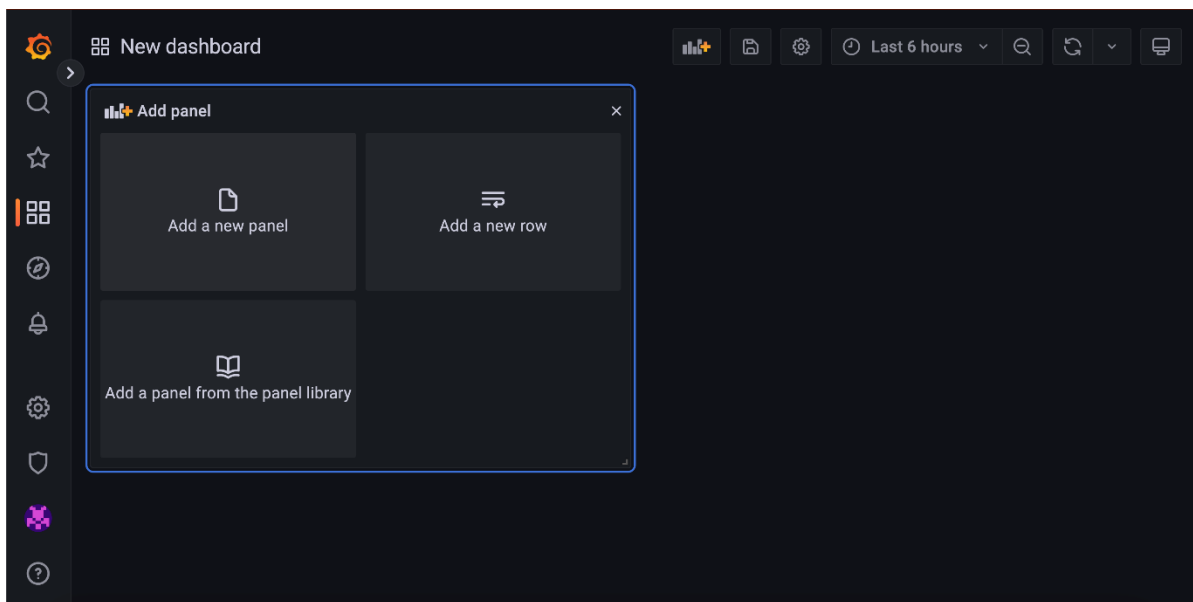


Рисунок 4.9

В налаштуваннях інтеграцій та підключень (рисунок 4.10) сконфігуруйте пункт з ClickHouse.

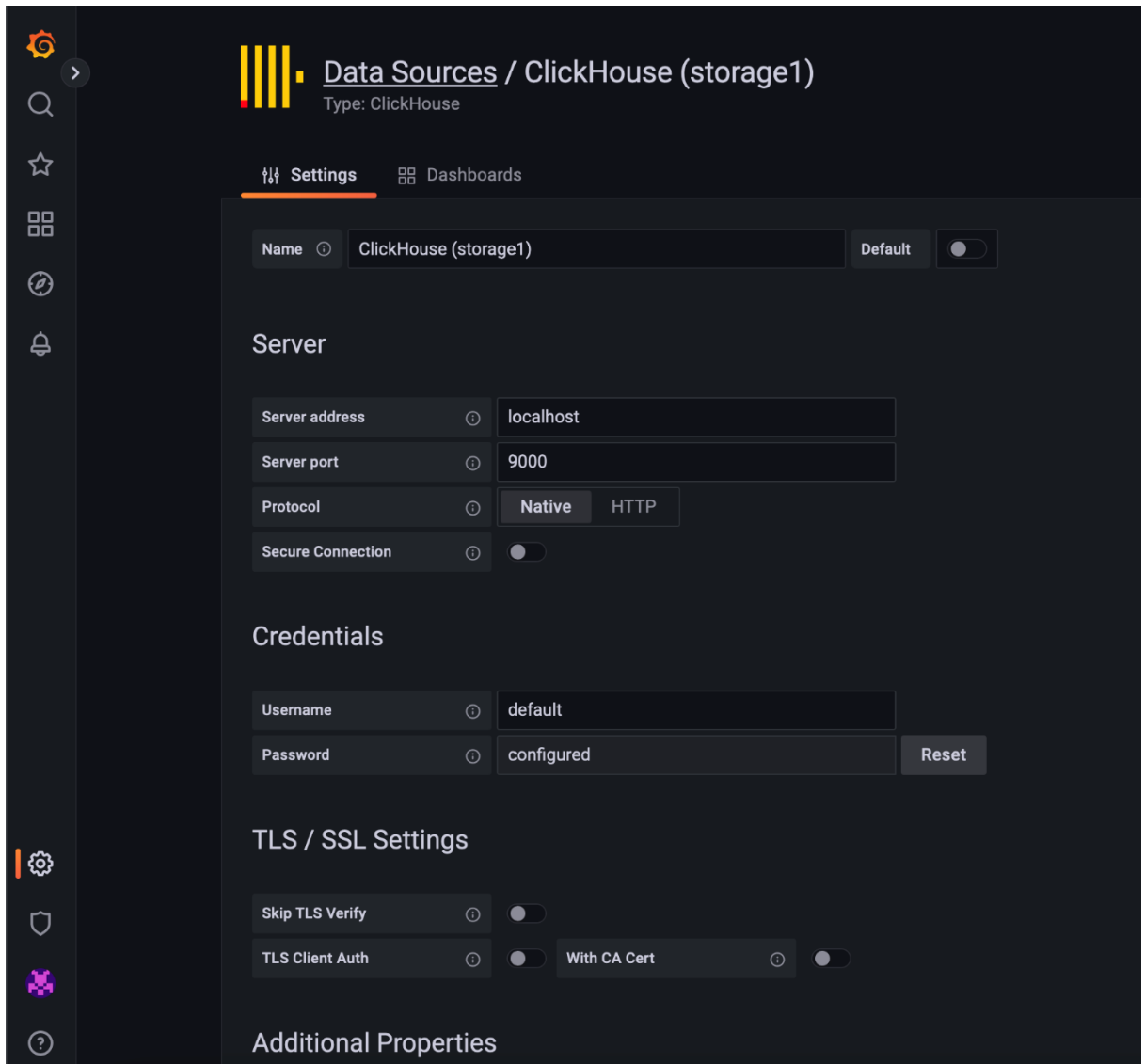


Рисунок 4.10

Далі створіть панель (рисунок 4.11), де буде сам графік.

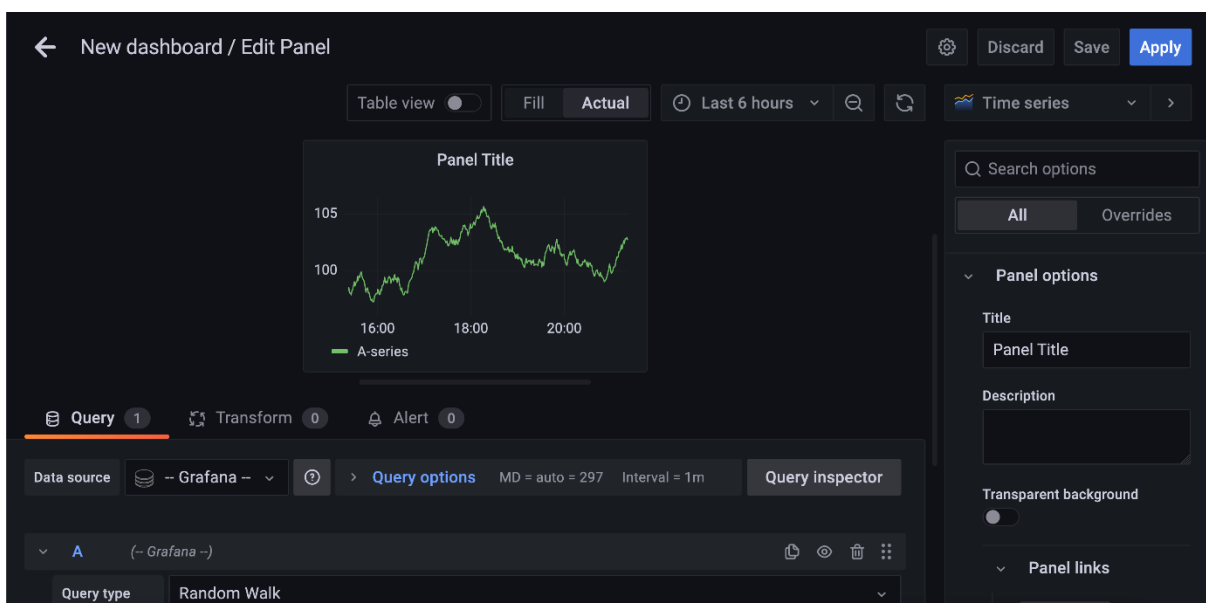


Рисунок 4.11

В пункті джерел (рисунок 4.12) оберіть ClickHouse.

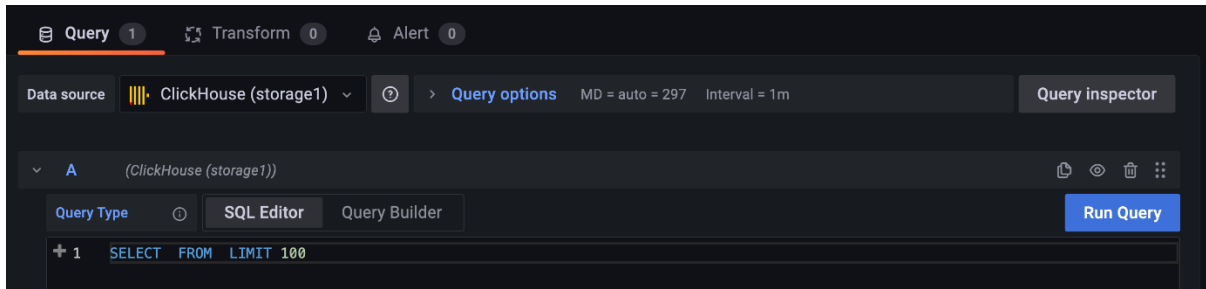


Рисунок 4.12

В поле для SQL запиту потрібно вставити наступний код:

SELECT

```
toStartOf${group}(greatest(broker1TradeTime, broker2TradeTime)) as time,
concat(broker1Symbol, ' > ', broker2Symbol) as symbol,
avg(spreadBidBid) as CLOSE_spread,
avg(spreadAskAsk) as OPEN_spread
```

FROM spreads.\${table}

WHERE

```
broker1TradeTime BETWEEN $__fromTime AND $__toTime
AND broker1Symbol = '${symbol_1}'
AND broker2Symbol = '${symbol_2}'
```

GROUP BY

```
time,
symbol
```

ORDER BY time

Збережіть цей запит, бо ще потрібно налаштувати динамічні змінні.

В налаштування дошки (рисунок 4.13) оберіть пункт зі змінними.

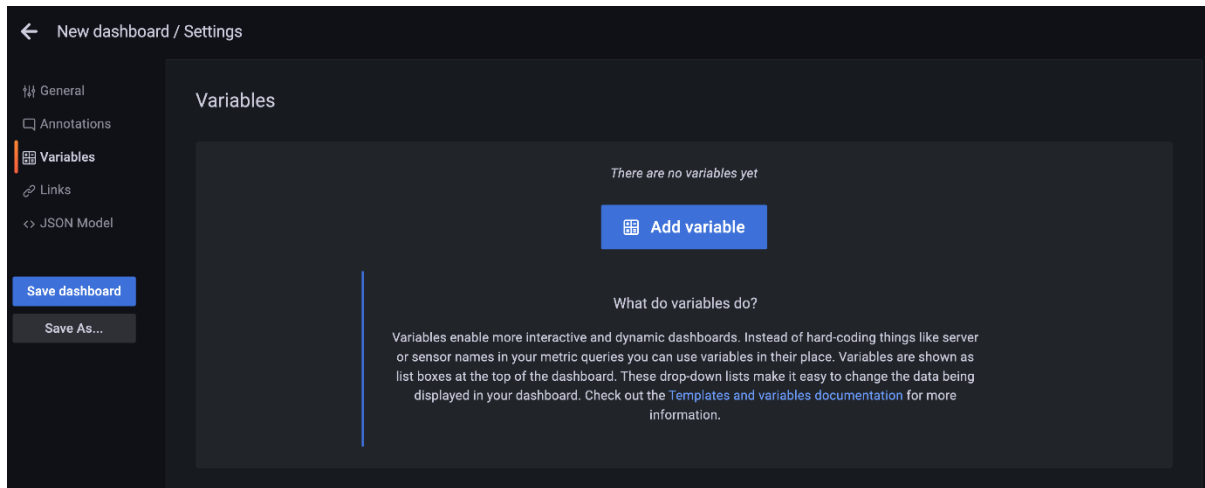


Рисунок 4.13

Додайте дві змінні (рисунок 4.14), які відповідають вибору символу:

– symbol_1 – код: `SELECT DISTINCT broker1Symbol FROM spreads.${table} WHERE broker1TradeTime BETWEEN $__fromTime AND $__toTime ;`

– symbol_2 – код: `SELECT DISTINCT broker2Symbol FROM spreads.${table} WHERE broker2TradeTime BETWEEN $__fromTime AND $__toTime ;`

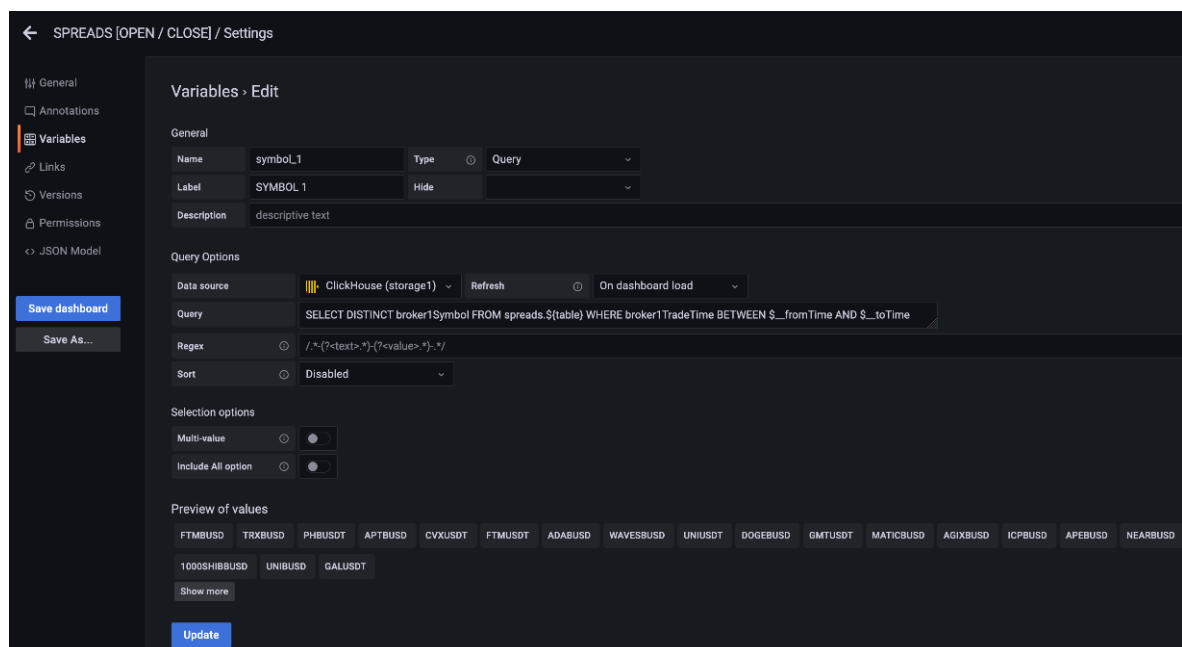


Рисунок 4.14

Додайте змінну table (рисунок 4.15) вибору таблиці з наступним запитом:
`SELECT name FROM system.tables WHERE database = 'spreads' AND name LIKE '%_spreads'`

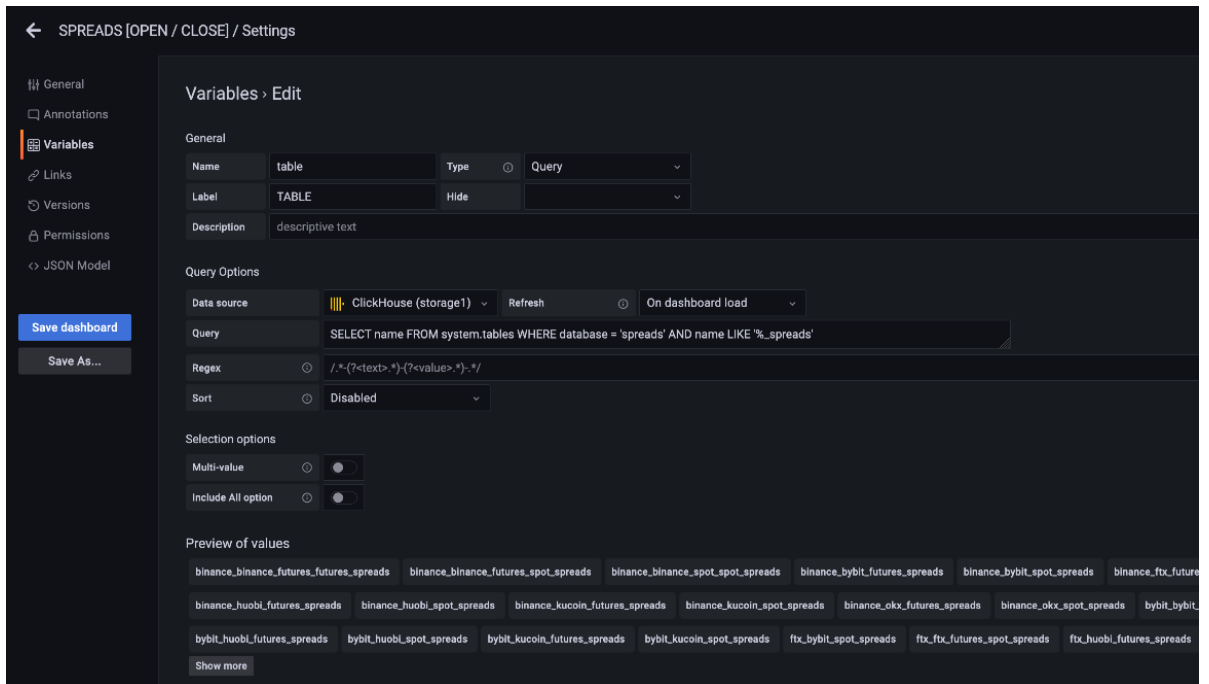


Рисунок 4.15

Додайте змінну (рисунок 4.16), яка буде відповідати за агрегацію.

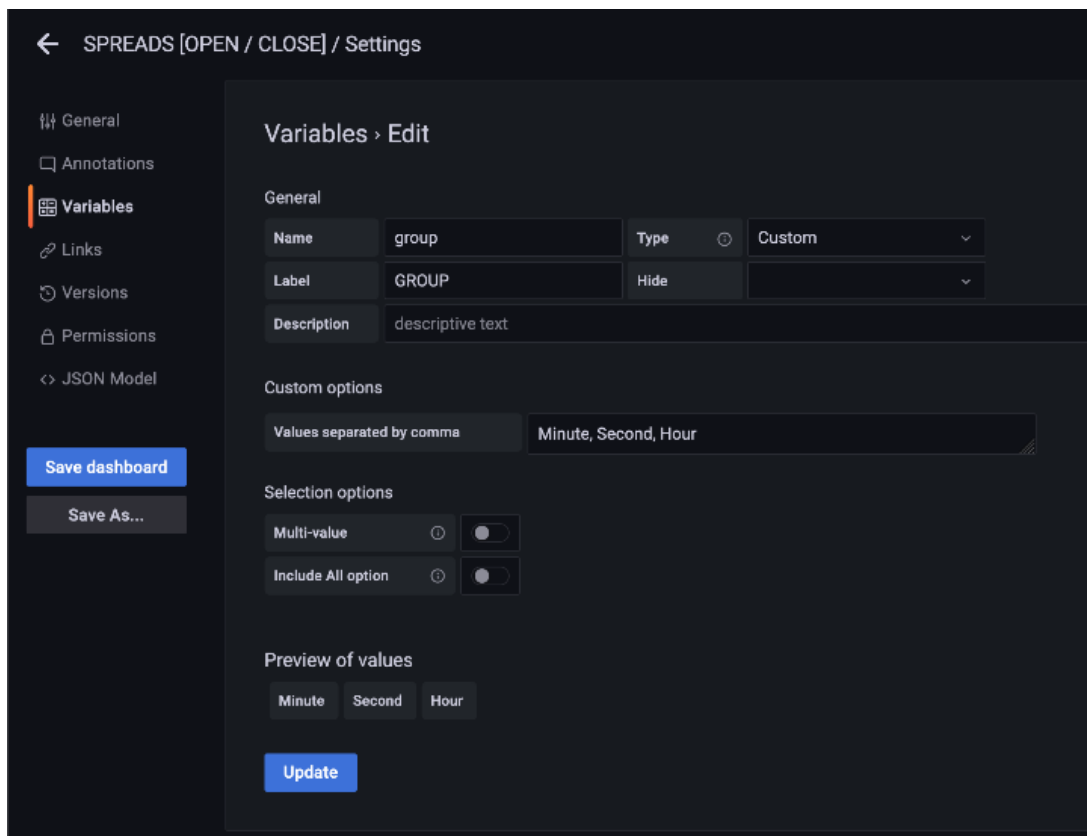


Рисунок 4.16

Далі потрібно знов зайти до панелі редагування графіка та зробити запит попередньо налаштувавши змінні, як вказано на рисунку 4.17.

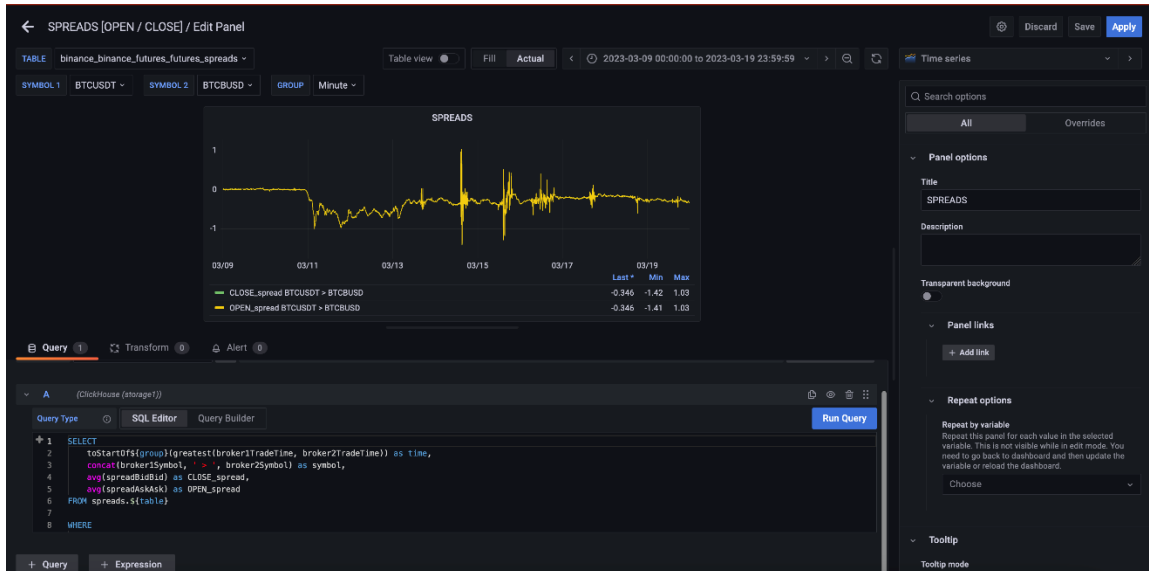


Рисунок 4.17

Інтерфейс повинен виглядати, як наведено в додатку А.

Висновки по розділу

Розроблено програмне та технічне забезпечення для реалізації системи збору, аналізу, зберігання та відображення даних з криптобірж.

Для програмного продукту висунуті вимоги, які необхідні для досягнення результату і задоволення потреб користувачів. Для розробки програмного продукту використані різноманітні засоби, які були необхідні для вирішення поставленої проблеми. Про їх переваги та недоліки також вказано.

Так як архітектура програмного забезпечення відіграє важливу роль у структуруванні програмного продукту і визначенні взаємозв'язків між його компонентами, то обрано мікросервісний підхід. Він допоможе створити такий продукт, який можна буде розширити під необхідні умови та задачі в майбутньому.

Для розподілу навантаження обрано популярний брокер повідомлень – RabbitMQ. Перевірена можливість до масштабування його інструментами.

ClickHouse обрано як система для асинхронного збереження даних. Так як Grafana має інтерфейс для роботи з ClickHouse, то її вибір був очевидний. Наведено приклади її використання. В інструкції користувача наведено повний алгоритм дій для налаштування.

Так як для роботи створено ще два сервіси, то в залежності від задач кожен був написаний різною мовою програмування:

- компонент обчислення спреду – Node.JS;
- компонент запису даних – C++.

Описана схема розроблених компонентів, які поєднуються один з одним.

Надана інструкція користувача, яка включає підготовку середовища для різних компонентів системи, встановлення необхідних компонентів та налаштування та запуск окремих сервісів.

Отже, виконана робота полягала в розробці програмного та технічного забезпечення для системи збору, аналізу, зберігання та відображення даних з криптобірж. Розроблені компоненти інтегруються з використанням наведених інструментів та технологій, а також надано докладні інструкції для налаштування та використання рішення.

ВИСНОВКИ

Перед вирішенням поставленої задачі спочатку зроблено огляд сучасних досліджень та рішень в області торгівлі криптовалютами для розуміння самої предметної області для подальшого дослідження. Оглянуто криптобіржі, проаналізовані їх відмінності від фондових. Щоб зрозуміти можливості інтеграції нашої системи розглянуті різні доступні торговельні інструменти: спотова торгівля, ф'ючерсні контракти, маржинальна торгівля, опціони.

Зроблено дослідження, щодо механізму укладання угоди між покупцем та продавцем: виставлення замовлень (ордерів) до ордербука; інша сторона, виставляючи таку ж саму ціну або краще, закриває угоду; після чого механізм криптобіржі нараховує учасникам криптовалюту чи контракти. Для ф'ючерсних контрактів після укладання угоди відкривається позиція, яка описує теоретичний його прибуток, можливість автоматичного закриття (ліквідація) та час експірації усіх угод, які відповідають сезонним контрактам.

Розглянуто популярні стратегії для торгівлі, кожна з них була частково проаналізована за критеріями використаного капіталу, ризиків та можливого прибутку. Для дослідження обрана торгівля за спредом, яка має незначні ризики та невеликі потенціальні прибутки з найбільшими вимоги до автоматизації – найперевірено ефективність її використання з урахуванням ризиків та можливостей до автоматизації. Для дослідження обраний наступний варіант торгівлі – ф'ючерсний контракт проти іншого ф'ючерсний контракту за однаковою базовою ціною.

Для аналізу параметру спреду розглянуто різні способи збору даних з криптобірж, перевірені можливості роботи з різними джерелами та протоколами для отримання подібної інформації. Оглянуто можливі дані, які необхідно обробляти та аналізувати для обчислення спреду.

В роботі висвітлені способи покращення обробки даних з криптобірж, зокрема застосування паралельної потокової обробки, агрегації та перетворення даних, використання брокерів повідомлень та систем асинхронного збереження

даних. Це може допомогти оптимізувати та прискорити обробку великих обсягів даних з криптобірж.

Особливу увагу приділено надійності збору даних, а також використанню брокерів повідомлень та систем асинхронного збереження даних для забезпечення стійкості та відновлюваності системи.

Описано архітектуру проекту, технології та компоненти, які повинні бути використані.

Перевіривши тестовий набір даних, результат показав ненадійні дані, які потрібно згладжувати для зменшення шуму та підвищення точності за рахунок менш інтенсивної торгівлі з кращими параметрами.

Досліджені три методи експоненційного згладжування, в яких не має сезонної залежності: Simple Exponential Smoothing, Holt's Linear Method, Additive Damped Trend. Перевірено результат і визначено, що для аналітичного відображення коефіцієнт α повинен бути менше 0.01 і отриманий результат майже однаковий, що виокремлює лише параметр швидкості для порівняння. Таким чином просте експоненційне згладжування має значну перевагу за швидкість серед усіх розглянутих методів.

Розроблено програмне та технічне забезпечення для реалізації системи збору, аналізу, зберігання та відображення даних з криптобірж.

Висунуті вимоги, які необхідні для досягнення результату і задоволення потреб користувачів та використані різноманітні засоби, які були необхідні для вирішення поставленої проблеми. Розглянуті їх переваги та недоліки, які вплинули на розробку архітектури проекту. Як результат – обрано мікросервісний підхід, який дає можливість розширити проект під необхідні умови та задачі в майбутньому.

Для розподілу навантаження обрано популярний брокер повідомлень – RabbitMQ. Перевірена можливість до масштабування його інструментами.

Для збереження даних – ClickHouse, який має можливість асинхронно зберігати великі масиви даних. Його інтеграція з Grafana дала можливість зробити зручний інтерфейс для користувачів.

За отриманою інформацією з інтерфейсу користувача зроблено декілька угод на криптобіржі, які згодом дали позитивний загальний результат, що є доказом, що така технологія збору інформації є ефективною.

Інтерфейс системи та результат експерименту наведені в додатку А.

Етапи роботи опубліковано у статті «Прогнозування короткочасних даних для крипторинку» [1].

ПЕРЕЛІК ПОСИЛАНЬ

1. Дуда В., Ролік О. Прогнозування короткочасних даних для крипторинку // Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління», 2023, № 1 (42), с. 141 – 152.
2. Дуда В.О., Терентьев О. М. Порівняння реалізацій систем збору інформації з криптобірж, у реальному часі, на мовах програмування C++, JavaScript та Python // Інформаційно-комунікаційні технології та сталий розвиток [колективна монографія за матеріалами XXI Міжнародної науково-практичної конференції (Київ, 14-16 листопада 2022 р.) / За заг. ред. С.О. Довгого]. – К.: ТОВ “Видавництво Юстон”, 2022., 58 – 62 с., ISBN 978-617-7854-76-9.
3. Muhammad Ashraf FAUZI, Norazha PAIMAN, Zarina OTHMAN, Bitcoin and Cryptocurrency: Challenges, Opportunities and Future Works. Journal of Asian Finance, Economics and Business Vol 7 No 8, 2020, pp. 695 – 704.
4. Rejeb, A., Rejeb, K., & Keogh, J. G, Cryptocurrencies in Modern Finance: a Literature Review. Etikonomi, 2021, pp. 93 – 118. DOI: 10.15408/etk.v20i1.16911.
5. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
6. Gandal N., Hamrick J.T., Moore T., Oberman T., Gandal N., Hamrick J.T., Moore T., Oberman T., Journal of Monetary Economics, 95, 2018 , pp. 86-96.
7. Eng-Tuck Cheah, John Fry, Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin, Economics Letters, Volume 130, 2015, pp. 32-36, ISSN 0165-1765, DOI: 10.1016/j.econlet.2015.02.029.
8. Kristoufek L, What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. PLoS ONE 10(4): e0123923, 2015, DOI: 10.1371/journal.pone.0123923.
9. Cevik, E.I., Gunay, S., Bagan, M.F. et al. The connectedness and risk spillovers between bitcoin spot and futures markets: evidence from intraday data. Ann Oper Res, 2022, DOI: 10.1007/s10479-022-04971-2.

10. Brière, M., Oosterlinck, K. & Szafarz, A. Virtual currency, tangible return: Portfolio diversification with bitcoin. *J Asset Manag* 16, 2015, pp. 365 – 373
DOI: 10.1057/jam.2015.5.
11. Galina Hale, Arvind Krishnamurthy, Marianna Kudlyak, and Patrick Shultz, How Futures Trading Changed Bitcoin Prices. FRBSF Economic Letter. Research from Federal Reserve Bank of San Francisco, 2018.
12. Aurélien Alfonsi, Alexander Schied, Alla Slynko, Order Book Resilience, Price Manipulation, and the Positive Portfolio Problem. *SIAM Journal on Financial Mathematics* 2012 3:1, pp. 511-533.
13. Joel Hasbrouck, Gideon Saar, Low-latency trading. *Journal of Financial Markets*, Volume 16, Issue 4, 2013, pp. 646-679, ISSN 1386-4181, DOI: 10.1016/j.finmar.2013.05.003.
14. Guilbaud, Fabien & Pham, Huyên. (2011). Optimal High Frequency Trading with Limit and Market Orders. *Quantitative Finance*. 13. DOI: 10.2139/ssrn.1871969.
15. Duarte, E., Trailing Stop-Loss and Re-Entry Strategies in Europe. *European Journal of Business and Management Research*, 7(3), 2022, pp. 118–123. DOI: 10.24018/ejbmr.2022.7.3.1426.
16. K. Jain, P. Jain and T. H. McInish: The Predictive Power of Limit Order Book for Future Volatility, Trade Price, and Speed of Trading. *SSRN Electronic Journal*, 2011.
17. Jiarui Ni and Chengqi Zhang: An Efficient Implementation of the Backtesting of Trading Strategies. *ISPA 2005: Parallel and Distributed Processing and Applications*, 2005, pp. 126–131.
18. Elie Bouri, Chi Keung Marco Lau, Tareq Saeed, Shixuan Wang, Yuqian Zhao, On the intraday return curves of Bitcoin: Predictability and trading opportunities, *International Review of Financial Analysis*, Volume 76, 2021, ISSN 1057-5219, DOI: 10.1016/j.irfa.2021.101784.
19. Amirzadeh, R.; Nazari, A.; Thiruvady, D. Applying Artificial Intelligence in Cryptocurrency Markets: A Survey. *Algorithms* 2022, 15, 428. DOI: 10.3390/a15110428.

20. T. Guo and N. Antulov-Fantulin: Predicting short-term Bitcoin price fluctuations from buy and sell orders, 2018.
21. Lubnau, Thorben, "Spread trading strategies in the crude oil futures market," Discussion Papers 353, European University Viadrina Frankfurt (Oder), Department of Business Administration and Economics, 2014.
22. Martin Papík, Assessing WebSocket Protocol Performance for Real-Time Cryptocurrency Algorithmic Trading with Compiled, Intermediate and Interpreted Programming Languages in Cloud Environment. School of Computing, Teesside University, Middlesbrough TS1 3BA, United Kingdom, 2020.
23. Dragoni, Nicola et al. "Microservices: Yesterday, Today, and Tomorrow". In: Present and Ulterior Software Engineering. Ed. by Manuel Mazzara and Bertrand Meyer. Cham: Springer International Publishing, 2017, p. 201. ISBN: 9783319674254. DOI: 10.1007/978-3-319-67425-4_12.
24. Wang, Cong & Wang, Qian & Ren, Kaili & Lou, Wenjing. (2009). Ensuring Data Storage Security in Cloud Computing.. IACR Cryptology ePrint Archive. 2009. 81. 10.1109/IWQoS.2009.5201385.
25. Lavasani, Mitra & Raeisi, Nahid & Sotudeh-Gharebagh, Rahmat & Zarghami, Reza & Abonyi, János & Mostoufi, Navid, Big data analytics opportunities for applications in process engineering. Reviews in Chemical Engineering. 39, 2021, DOI: 10.1515/revce-2020-0054.
26. Takashi Kanamura, Svetlozar T Rachev and Frank J Fabozzi – "A Profit Model for Spread Trading with an Application to Energy Futures", 2010, DOI:10.1186/s40854-021-00321-6.
27. Fang, F., Ventre, C., Basios, M. et al. Cryptocurrency trading: a comprehensive survey. Financ Innov 8, 13, 2022, DOI:10.1186/s40854-021-00321-6.
28. Trevethan R, Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice. Front. Public Health 5:307, 2017, DOI: 10.3389/fpubh.2017.00307.

29. Hyndman, Rob J., and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.
30. A. Supriatna, D. Susanti, E. Hertini. Application of Holt exponential smoothing and ARIMA method for data population in West Java. *IOP Conf. Series: Materials Science and Engineering* 166, 2017, 012034 DOI: 10.1088/1757-899X/166/1/012034.
31. Charles C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages, *International Journal of Forecasting*, Volume 20, Issue 1, 2004, pp. 5-10, ISSN 0169-2070, DOI: 10.1016/j.ijforecast.2003.09.015.
32. Giacomo Sbrana, Andrea Silvestrini, Forecasting with the damped trend model using the structural approach, *International Journal of Production Economics*, Volume 226, 2020, 107654, ISSN 0925-5273, DOI: 10.1016/j.ijpe.2020.107654.

ДОДАТОК А

Графічний матеріал