

Fundamentals of Arbitrage

HOW DOES IT WORKS?
HETHA.IO TEAM

Table of Contents

INTRODUCTION	2
1. FUNDAMENTALS OF ARBITRAGE. MAIN CHALLENGES AND WAYS TO ADDRESS THEM	
2. FUNDAMENTALS OF ARBITRAGE. FUNDAMENTAL CONCEPTS OF HETHA.IO ARBITRAGE, WHAT CHAINS ARE, THEIR TYPES, EXAMPLES OF APPLICATIONS	10
3. HETHA.IO. CONCEPT OF DISTRIBUTED ARBITRAGE	13
4. HETHA.IO. ARCHITECTURE	16
5. RISKS VS PROFITABILITY. HOW TO INCREASE THE PROFITABILITY OF AN ARBITRAGE DEAL. TRENDING, DISCOUNTS.	18
6. PORTFOLIO. HOW TO EVALUATE REAL PROFITABILITY	25
7. PORTFOLIO. CREATING A PORTFOLIO.....	27
9. TRADING BOT. SETTINGS OVERVIEW	34
11. TRADING BOT. MANAGING THE TRADING BOT, FINE-TUNING SETTINGS, AND CHECKING OPERATION	40
12. TRADING BOT. DEVELOPING ADDITIONS.....	44
13. TRADING BOT. API, FEEDBACK WITH GUI.....	46
14. PERSONAL ACCOUNT. BASIC DIFFERENCES IN TARIFF PLANS.....	48
15. PERSONAL ACCOUNT. «MARKETS» SECTION.....	49
16. PERSONAL ACCOUNT. «STATISTICS» SECTION.....	60
17. PERSONAL ACCOUNT. «SERVER» SECTION	64
18. PERSONAL ACCOUNT. OTHER SECTIONS	75
APPLICATION	76

INTRODUCTION

Arbitrage in economics can be spatial, temporal, equivalent, interest-based, and so on. In this book, we will focus on only one type of arbitrage — spatial arbitrage. It refers to generating profit from price differences on identical or related assets at the same time. Arbitrage transactions themselves are not merely speculative; they are the primary mechanism for price alignment in global markets. Therefore, arbitrage transactions are not only profitable but also essential for market functioning. They are what make the markets for cryptocurrencies, currencies, and commodities mature and balanced. Typically, trading platforms have their own arbitrage mechanisms, but this does not prevent external services from profiting as well.

1. FUNDAMENTALS OF ARBITRAGE. MAIN CHALLENGES AND WAYS TO ADDRESS THEM

How does it work?

Suppose that on the Binance exchange, the BTC/USD pair is trading at \$65,000 per Bitcoin. At the same time, on the Bitfinex exchange, the BTC/USD pair is trading at \$65,500. If we buy Bitcoin on Binance for \$65,000 and simultaneously sell one Bitcoin on Bitfinex for \$65,500, we can earn \$500.

- Binance BTC/USD buy 1 BTC – \$65,000
- Bitfinex BTC/USD sell 1 BTC – \$65,500

This is an example of a simple cryptocurrency arbitrage trade. As shown in the example, an arbitrage trade differs from a speculative one in that orders are placed in both directions, minimizing the risk of losses. By buying and selling simultaneously, the trader does not risk their trading portfolio, as no open orders remain after the arbitrage trade is completed. Consequently, there is no need to worry about market direction.

While it seems straightforward, logical, and profitable, performing such transactions comes with several challenges and problems, such as:

- Identifying arbitrage opportunities
- Managing the trading portfolio and the associated capital dilution
- Costs of executing trading operations
- Unclosed legs of arbitrage trades
- Asset migration

These are the most common problems. Other issues exist but are more specific and will be discussed in future articles.

Identifying Arbitrage Opportunities

The speed at which events occur on cryptocurrency exchanges almost immediately rules out the possibility of manually searching for and processing these events. In actively traded pairs, events (ticks) can occur at frequencies ranging from several per minute to dozens per second. This dynamic nature makes manual arbitrage trading ineffective, as each trade requires completing the following steps:

- Continuous monitoring (of at least two exchanges and one pair)
- Detecting an arbitrage event
- Executing the first order
- Executing the second order

Monitoring requires constant observation of exchange rates since arbitrage events, especially in popular pairs, are rare. They may occur once a day or even once a month, and their frequency cannot be predicted as it depends heavily on market activity.

In this case, using automated scripts for detection is much more efficient. Such scripts can scan dozens of exchanges with hundreds of cryptocurrency pairs.

Additionally, these scanners can identify complex arbitrage chains, such as:

- Binance BTC/USD buy
- KuCoin ETH/BTC buy
- Bitfinex ETH/USDT sell

This significantly increases the profitability of the arbitrage business. Generally, multi-leg arbitrage opportunities are more lucrative than two-leg ones. However, the more legs an arbitrage chain has, the harder they are to close, increasing the risks.

A well-designed arbitrage chain scanner should identify opportunities within fractions of a second. Otherwise, given the speed of ticks, arbitrage opportunities may no longer be relevant within seconds. In addition to fast detection, it's also crucial to ensure the speed of order execution. Automated bots typically execute orders via REST API, which is somewhat slower than FIX or WebSocket connections. Although WebSocket connections offer theoretical advantages, they are not standardized, complicating implementation. Nevertheless, REST API connections provide satisfactory results. In automated arbitrage trading systems, it's entirely feasible to execute orders within one second from the tick that triggered the arbitrage opportunity. This ensures modern arbitrage systems can effectively process events as needed, making them preferable for this field.

Scanners can simultaneously work with thousands of pairs across various exchanges. However, this creates a new problem: the large volume of information requires filtering mechanisms to identify high-profit pairs. When selecting pairs for further work, attention should be given to several aspects:

- Profitability generation
- Availability of reverse opportunities

Profitability is relatively straightforward: pairs with the highest profit percentages should be prioritized. Reverse opportunities, however, may raise questions.

For example, suppose arbitrage opportunities occasionally arise between Binance and KuCoin:

- Binance BTC/USD buy
- KuCoin BTC/USD sell

At the start of trading, both exchanges have 0.1 BTC and \$5,000 in their accounts. As arbitrage opportunities are executed, assets will gradually separate: one exchange will accumulate BTC, while the other will accumulate USD. Eventually, Binance will have only BTC (since it is actively being purchased), and KuCoin will have only USD. At this point, arbitrage trading will come to a halt.

To resume trading, balances must be rebalanced. This means transferring half of the BTC balance from Binance to KuCoin and half of the USD balance in the opposite direction. After rebalancing, arbitrage trading can resume.

However, transferring cryptocurrencies like BTC, and especially fiat currencies like USD, incurs significant costs. Additionally, USD transfers require using the banking system.

The most advantageous rebalancing option is reverse trades. These trades do not need to generate significant profits—breaking even is sufficient. Furthermore, reverse trades of varying profitability can offset the costs of rebalancing assets via transfers. This is why it's essential to analyze the history of arbitrage events for the presence of reverse opportunities.

Of course, such analysis can be performed by specialized bots. However, there are nuances that may pose challenges. This primarily concerns recurring arbitrage events. Regularity with a bias toward one direction—frequent generation of similar events with the same pairs, exchanges, and buy-sell directions—is an important signal for filtering pairs.

To detect such biases, the following exchange parameters should be checked:

- Liquidity on the exchange
- Withdrawal capabilities

These parameters can create significant price discrepancies. For example, the SOL/USDT pair on Binance compared to CEX shows a 5% difference. While this appears to be a great arbitrage opportunity, further investigation reveals that CEX lacks liquidity and has extremely low trading volumes. This increases the risk that the order placed on this exchange won't close.

Another example involves the WRX/USD pair on Binance and KuCoin. Arbitrage profits may range from 8–9%. However, an investigation reveals that while Binance allows WRX withdrawals, KuCoin has restrictions on WRX deposits and withdrawals. This means that as arbitrage trades are executed, WRX will accumulate on Binance, making balance redistribution difficult due to KuCoin's limitations.

Portfolio Formation Approaches

Portfolio formation can follow either a profitability-focused approach or a portfolio-based approach.

1. **Profitability-Focused Approach:** This approach requires searching for and analyzing events to form a high-profit portfolio.
2. **Portfolio-Based Approach:** Here, the portfolio is formed based on subjective assessments of its growth potential as an investment model, with arbitrage serving as passive income for the portfolio. In this case, the previously mentioned rules do not apply.

Trading Portfolio and Associated Capital Dilution

Profitability-Focused Approach

The classical, old-school arbitrage approach involves holding all tradable pairs in the portfolio, as trading takes place in the SPOT market. This means that after identifying profitable pairs, funds must be allocated across the relevant exchanges and pairs defined in the arbitrage opportunities. For instance, after analyzing a stream of arbitrage events, we select three exchanges: Binance, Bitfinex, and KuCoin. The pairs of interest are KEY/BTC, BAND/BTC, and TFUEL/BTC. The initial balance is 1 BTC. According to the classical approach, the initial balance needs to be divided among the currencies KEY, BAND, TFUEL, and BTC. After splitting the balance, trading can commence.

Despite its higher profitability, this approach carries certain risks:

- Rare pairs generally lack sufficient liquidity, posing a significant risk of failing to close orders during arbitrage events.
- Rare pairs are often subject to Pump & Dump schemes. While arbitrage transactions may yield high profits, there is a risk that the nominal portfolio value may drop, which might not be offset by arbitrage gains.

A more advanced approach involves the use of Margin Trading, where portfolio formation is unnecessary. Currencies like KEY, BAND, TFUEL, and BTC can be borrowed from the respective exchanges when placing orders. Margin trading involves leveraging, which increases profitability.

However, higher profitability also brings increased risks: if one leg of an arbitrage event fails to close, it can impact the deposit, potentially leading to a Margin Call.

Margin trading is undoubtedly more convenient than the classical approach. On the one hand, it eliminates the risks of including underperforming cryptocurrencies in the portfolio, but on the other hand, it requires strict adherence to Money Management rules.

Portfolio-Based Approach

In this approach, the classical portfolio-building model is followed, disregarding arbitrage revenue analysis. The key factor in portfolio construction is its investment potential. In other words, the portfolio should include cryptocurrencies with growth potential. Arbitrage income is secondary in this approach.

For example, a portfolio might include BTC, ETH, SOL, and USDT. Arbitrage income can still be expected, as arbitrage opportunities occur across all pairs without exception.

With this approach, the main source of earnings is the portfolio's value itself, while arbitrage provides passive income. By operating within one's own portfolio, the risks of losses are almost negligible.

Costs of Performing Trading Operations

Like any activity, arbitrage incurs its own costs:

- Initial portfolio allocation
- Exchange fees
- Balance equalization through inter-exchange transfers

Initial Portfolio Allocation

Typically, assets are initially held in a single currency. Allocating the initial portfolio requires splitting the starting deposit among the exchanges planned for arbitrage. For example, if the initial deposit is in BTC and four exchanges are to be used, four transfers must be made.

It is important to account for transfer fees. For large volumes, these fees are negligible, but for small amounts, they can be significant. For instance, four transfers might cost approximately \$50.

Next, the base asset must be exchanged for 3–4 portfolio currencies. Each exchange incurs a fee of about 0.1%–0.3% of the transaction amount.

While initial allocation requires investment, passive income from arbitrage events can compensate for these costs.

Exchange Fees

Each exchange charges a commission for trading operations, which is their primary source of revenue. Depending on the exchange, fees can vary significantly but typically range between 0.1% and 0.3% of the transaction amount.

Arbitrage events must always account for these exchange fees. For example, conducting an arbitrage transaction with 1 BTC at a price of \$70,000 on Binance and Bitfinex would require paying \$210 in fees:

$$\$70,000 * 0.1\% \text{ (Binance)} + \$70,000 * 0.2\% \text{ (Bitfinex)} = \$70 + \$140 = \$210$$

Thus, buying 1 BTC on Binance for \$70,000 and selling it on Bitfinex for \$70,210 would result in a **0% profit**.

Balance Equalization Through Inter-Exchange Transfers

As mentioned earlier, when assets become heavily separated across exchanges, it becomes necessary to move them. While reverse trades can serve as an alternative to transfers, if they fail, asset migration may be the only viable solution to equalize balances.

The cost of transfers varies greatly depending on the type of asset and the exchanges involved. Bitcoin transfers are among the most expensive, costing \$10–\$15, followed by Ethereum (\$5–\$8) and ERC20 tokens. Other assets are significantly cheaper to transfer (\$0.5–\$1).

Such fees are particularly burdensome for small-scale operations. This is a common challenge in arbitrage, where the size of operational assets is critical. Larger assets generate higher turnover, which in turn qualifies for discounts that can significantly improve overall profitability. Discounts will be discussed later.

Unclosed Legs of an Arbitrage Trade

The primary risk in arbitrage lies in unclosed legs. The "legs" of an arbitrage event refer to the orders that constitute it. Is it possible to guarantee a 100% closure of arbitrage trade legs? To answer this, we need to examine the types of orders used in arbitrage processing.

Markets allow for two types of orders: Market Orders and Limit Orders. The difference between them is that a Market Order guarantees the execution of the order but not the price, whereas a Limit Order guarantees the price but not the time of execution.

Can Market Orders be used to guarantee 100% closure of trades? Sometimes, but not always. The problem lies in the floating nature of the price at which a Market Order will execute. This means that the price difference calculated during the arbitrage event analysis cannot be guaranteed. This is particularly critical when the profitability of an arbitrage event is a fraction of a percent (typical for a single arbitrage transaction). Events with profits exceeding 1% are rare, especially for top-tier cryptocurrencies. Thus, relying on Market Orders is risky.

Limit Orders, on the other hand, provide the ability to specify an exact price, ensuring that when all legs are closed, the arbitrage trade will be profitable within the calculated model. However, there is no guarantee that the legs of the trade will close.

This creates a complex but not hopeless situation. A key way to increase the likelihood of closing arbitrage legs is to leverage exchange commission discounts. Many platforms offer flexible discount systems. Some of these discounts can be applied to adjust the price, allowing orders to be placed slightly below or above the declared price in the arbitrage chain. This significantly increases the chances of order execution.

Asset Migration

This issue exists only within the classical, conservative model of arbitrage trading. In the advanced model, where trading is conducted with borrowed assets for specific operations, this problem does not arise. However, as previously mentioned, this method introduces other challenges related to the use of borrowed funds.

The essence of the problem is that as arbitrage trades are conducted, there is a constant inflow and outflow of various assets on different exchanges. Over time, this migration reduces the efficiency of arbitrage. A shortage of assets arises: funds needed for certain arbitrage events may be unavailable

because they are held on other exchanges or in pairs not currently required. This leads to missed opportunities for executing profitable and high-quality arbitrage events.

One solution to this problem is reverse trades, which will be discussed below. The most radical solution is inter-exchange fund transfers. However, as previously mentioned, this entails additional costs.

Arbitrage trading itself is not a source of abundance. It is a meticulous process requiring attention, decision-making, and analysis. Unlike speculative trading, arbitrage carries fewer risks, and various methods exist to optimize and mitigate those risks.

As with any other type of business, arbitrage adheres to the principle of risk-to-reward ratio. As risks increase, so do potential profits, and vice versa. Below, we will discuss automated solutions for generating arbitrage events and mechanisms to reduce trading risks and increase profitability.

2. FUNDAMENTALS OF ARBITRAGE. FUNDAMENTAL CONCEPTS OF HETHA.IO

ARBITRAGE, WHAT CHAINS ARE, THEIR TYPES, AND EXAMPLES OF APPLICATIONS

In the context of the HETHA.IO arbitrage service, an arbitrage event is represented as an arbitrage chain, which is the core data stream generated by HETHA.IO.

An arbitrage chain refers to a sequence of exchange nodes that must be executed immediately and in parallel to achieve a specified profit.

Example of an arbitrage chain:

bfx:btc/usd-bfx:eth/usd-bnc:eth/btc

Here, the hyphen (-) separates individual nodes, and each node consists of the abbreviated exchange name and the trading pair.

Each chain has a base currency, which is the currency used to calculate the profit of the chain. The base currency can only be USD or BTC. It is worth noting that USD actually represents USDT in the chains, though this is not explicitly reflected. Initially, exchanges used either USD or USDT depending on liquidity. Today, USD is universally USDT on all supported exchanges due to its easier migration compared to fiat currency.

Profit Calculation for Chains:

- If the base currency is USDT, the calculation starts with 100 USDT.
- The first node performs the conversion, deducting the respective exchange commission.
- The result is passed to the second node, where another conversion occurs, accounting for the next commission.
- Similarly, the output of the second node is fed into the third node, where the final calculation is made, considering the commission.
- The final operation in the chain ensures the result is in the same base currency.

For example, starting with 100 USDT, the output might be 100.05 USDT, indicating a profit margin of 0.05%. All commissions are factored into this calculation.

A similar method applies to BTC, with an equivalent value of 100 USDT. Examples of Chains:

- kcn:ada/usd-bfx:ada/usd — Depth: 2, Base Currency: USDT
- bfx:trx/btc-kcn:trx/btc — Depth: 2, Base Currency: BTC
- bfx:btc/usd-bfx:eth/usd-bnc:eth/btc — Depth: 3, Base Currency: BTC

Each chain includes details such as profit, generation time, type, and more. Chain profitability typically does not exceed **1%**, though exceptions occur.

Generation time refers to the interval from the last incoming tick in the chain to its receipt on a remote server. This is usually less than 1 second, averaging between 0.1–0.5 seconds, depending on market conditions.

- **Calm markets:** Tick rate <150,000 per minute ensures generation time <1 second.
- **Volatile markets:** Tick rate >150,000 per minute may cause delays and render chains outdated due to rapid price changes.

It is recommended to discard chains with a generation time exceeding 1 second as they are less reliable and prone to market volatility risks.

HETHA.IO operates across 6 markets, processes 3,000 pairs, monitors 30,000 chains in parallel, and tracks 100,000–300,000 ticks per minute. The number of profitable chains (yielding >0.02%) can reach 10,000 per hour during high activity but averages around 500 per hour in stable markets. Millions of other chains are discarded as unprofitable.

Chains can be classified by:

- Depth
- Full or Short
- Forced (Reversed)
- Filtered

Chain Depth

Chains can have depths of **2** or **3**. Depth **4** chains were considered previously but were abandoned due to inefficiency:

1. Exponentially increased server load.
2. Profits were not significantly higher than depth-3 chains.

Depth-2 Chains - classic arbitrage. An asset is purchased on one exchange with a specific currency and sold on another exchange at a higher price in the same currency. Input and output currencies are identical.

Example:

- kcn:ada/usd-bfx:ada/usd

Depth-3 Chains - these chains involve different purchase and sale currencies, requiring a third node to finalize the profit.

Example:

- Initial chain: bnc:eth/usd-bfx:eth/btc
- Final chain: bnc:eth/usd-bfx:eth/btc-bnc:btc/usd

Longer chains increase the risk of failure. To mitigate this, it is recommended to increase the required initial profitability by **1.5x** to compensate for potential risks.

Full and Lite Chains

First, it's essential to explain the format used for storing order book prices in the HETHA.IO system. Since the service processes changes across thousands of orderbooks in parallel and analyzes all changes when building chains, the orderbooks are stored in an optimized format as follows:

```
{
  a: [[price_1, vol_1]],
  aW: [price_1],
  b: [[price_1, vol_1], [price_2, vol_2], [price_3, vol_3]],
  bW: [(price_1 * vol_1 + price_2 * vol_2 + price_3 * vol_3) / 3]
}
```

Here a - Ask prices, b - Bid prices, aW - Weighted average ask price, bW: Weighted average bid price. aW and bW represent weighted average prices for an order volume of \$5,000.

The size of the arrays depends on the position volumes for each price. The total volume must not exceed the equivalent of \$5,000. Any amounts exceeding this limit are discarded.

Lite chains are calculated using bB and aB prices:

- bB equals b[0] (or price_1 in array b).
- aB equals a[0] (or price_1 in array a).

These are the closest prices in the orderbook, forming the spread. Lite chains do not guarantee volume and may sometimes include chains with minimal volumes. Therefore, placing large volumes on Lite arbitrage chains is not recommended.

Full chains are calculated using aW and bW prices, which means they can handle larger volumes.

Forced Chains

As mentioned earlier, asset migration is a significant challenge for arbitrage services. If assets are heavily segregated across exchanges, arbitrage operations may halt. This occurs when asset locations do not match the requirements of the chains.

One automatic solution to this issue is reverse trades. Reverse trades help move funds back in the opposite direction, avoiding critical asset separation. Such chains may appear in the stream but often fail to pass profitability filters. To avoid this, settings should be configured to automatically detect asset separation. When a suitable trade is identified but has a low-profit percentage, the trade is executed without profit. This type of chain is called a forced chain.

Filtered Chains

Chain filtering is a loosely defined concept. The HETHA.IO service uses its own filtering mechanism, but this procedure is subjective. To ensure flexibility, users can adjust filtering parameters as they receive unfiltered chains and set their own filters to balance profit&risk. It's important to note that even minor parameter changes can lead to entirely different analytical results.

Standard Filtering Procedure Parameters:

Low Profit. Chains that do not meet the profitability criterion are rejected. (*Default: lite_chain – 0.1%, full_chain – 0.05%, force_chain – 0%*)

Long Chain Generation Time. If chain generation exceeds the threshold, it may no longer be relevant. (*Default: 1 second*)

Excessive Profit. Very high profit may indicate anomalies in chain generation, such as frozen order books, price distortions on exchanges, or other inconsistencies. (*Default: 10%*)

Exchange Provider Access. Chains are rejected if data from exchanges is unavailable or connections to exchanges are slow.

Price Deviation Check. In addition to chain data, real-time order book information is also transmitted. During filtering, the system compares order book prices with chain prices. If the deviation exceeds the threshold, the chain is rejected. (*Default: lite_chain – 0.05%, full_chain – 0.1%*)

Adjusting these parameters can significantly increase the number of arbitrage deals and profitability. This will be discussed in more detail later.

3. HETHA.IO. CONCEPT OF DISTRIBUTED ARBITRAGE

Scanning and analyzing markets is a complex task requiring significant computational resources to process data streams quickly. The system's responsibilities can be divided into three subtasks:

- Collecting exchange information
- Chain recalculation
- Server manager

All subtasks run in parallel and do not require deep integration, enabling them to operate on different hardware connected via a network.

The task of collecting exchange information is handled by a group of independent servers. Currently, there are 11 servers, but this number can easily be scaled. Any computing hardware with multiple cores and several gigabytes of RAM can be used as an exchange information collector. The group of collector servers is managed by a server manager, whose responsibilities include:

1. Distributing the orderbooks to be processed evenly across the collector servers.

2. Periodically reviewing markets to find new pairs and add them to the processing pipeline.

The complexity of processing ticks depends on the specific exchange. Some exchanges provide an order stream that requires manual construction of order books, while others supply ready-made order books. Processing ready-made order books and trimming them to the required volume is relatively straightforward (volume information can be found in the Full / Lite Chains section). However, handling and removing orders to construct order books is resource-intensive. Each entry into an order book triggers chain recalculations. The challenge lies not in assembling an order book from incoming orders but in avoiding unnecessary chain recalculations.

Ticks undergo rigorous filtering before being deemed relevant and requiring chain recalculations. For instance:

- If an incoming order does not change aW, bW, aB, or bB parameters (explained in the Full / Lite Chains section), the tick is ignored and not sent for recalculation.
- If a tick modifies the order book but the deviation is less than 0.01%, it is also ignored, as such a change is negligible even compared to exchange commissions.

It is important to note that each chain accounts for exchange commissions at every node. This means that at least 0.2% in a depth-2 chain and 0.3% in a depth-3 chain are allocated to cover commissions. Therefore, price changes of a few hundredths of a percent can be disregarded. Despite filtering, the service processes 100,000–300,000 ticks per minute, highlighting the necessity of filtering as a forced measure.

Depending on the workload planned by the server manager, a separate process is assigned to maintain each order book. This process establishes its own WebSocket connection with the exchange and the chain calculation system. The server manager monitors these processes and terminates or restarts them as needed. Hundreds of processes handle ticks in parallel, forwarding filtered data for chain recalculation.

There is no limit to the number of collector servers. Each server can handle its assigned workload, and the server manager can redistribute the load when necessary.

Maintaining orderbooks and filtering ticks is the most resource-intensive part of the system. While chain recalculation requires fewer servers, it demands specialized hardware.

Chain Recalculation (GPU)

Orderbooks streams from collector servers, whose ticks have passed filtering, are sent to the calculation server. Chain recalculation is more centralized, requiring significantly fewer servers than data collection. This is because the data stream is architecturally compressed and concentrated at a single point, which then distributes it to client servers via WebSocket.

The core of the calculation server is a GPU, where massive recalculations of order books are performed to determine the current profitability of chains. Out of millions of chains recalculated every hour, only 50–100 are profitable, while the rest are discarded after processing.

Since the number of orderbooks sent to the GPU exceeds 5,000, it is crucial to use GPUs with a high number of threads for parallel computations. Additionally, the GPU must have a large amount of RAM to ensure all calculations are performed entirely within the GPU, avoiding external memory access. This is the only way to achieve the computation speed necessary to process data in tenths or even hundredths of a second.

In parallel with recalculating and outputting profitable chains, the calculation server forwards the modified order books. This places a load on both the GPU cores and the network connections of the calculation server. Currently, a single calculation server is operational, with its GPU utilization ranging between 10–15% of its total computational capacity. As the system approaches its threshold, additional calculation servers will be brought online.

Server Manager

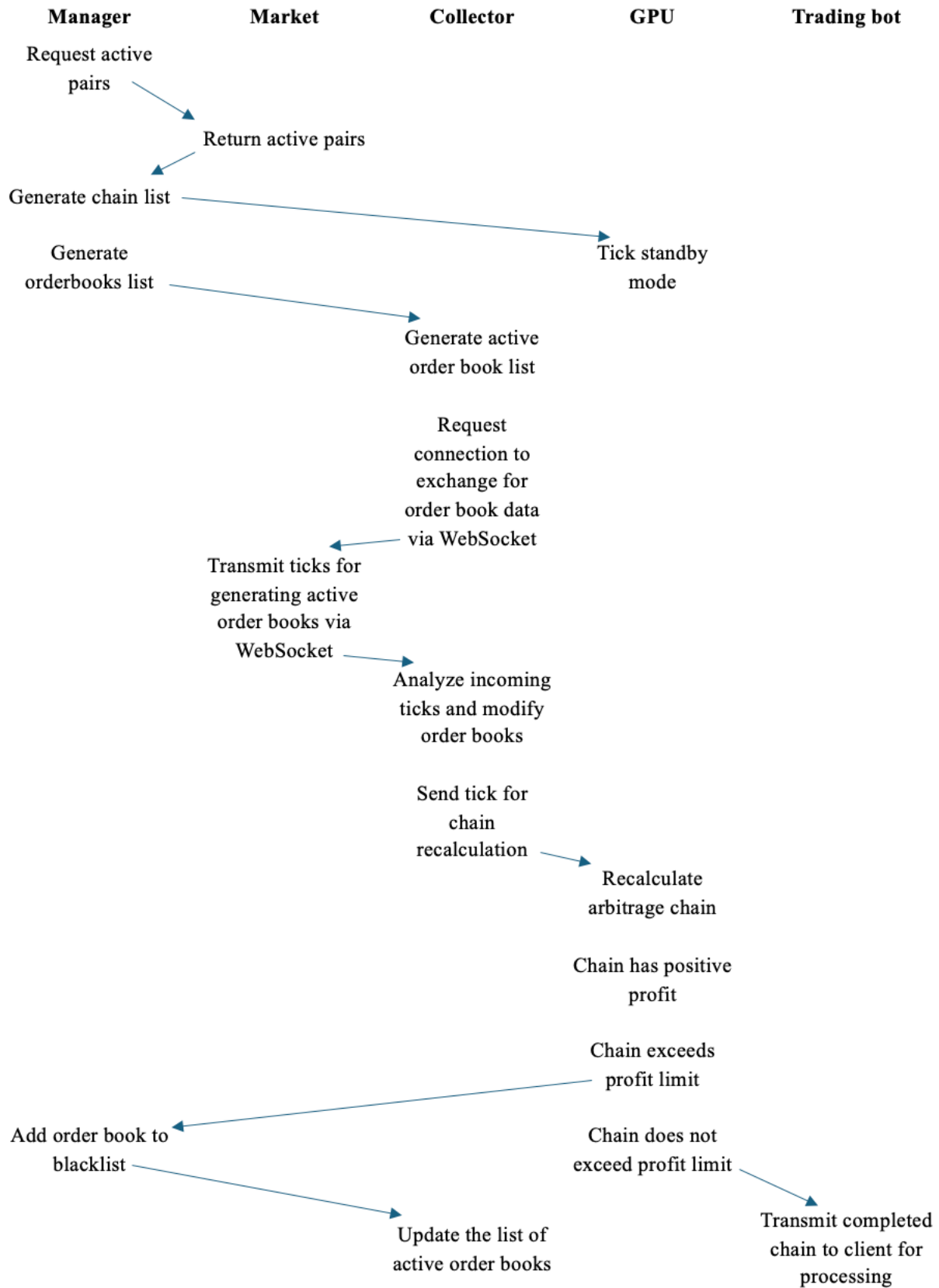
The server manager is responsible for distributing the workload across the collector servers. It also monitors changes occurring on trading platforms, particularly changes in the list of active pairs, as previously mentioned.

Another key task of the server manager is controlling the results of chain recalculations. Occasionally, different cryptocurrencies with the same name but varying rates are traded on the market. This can result in non-existent chains with incorrect profit calculations. The server manager filters such chains and automatically blacklists them for further investigation.

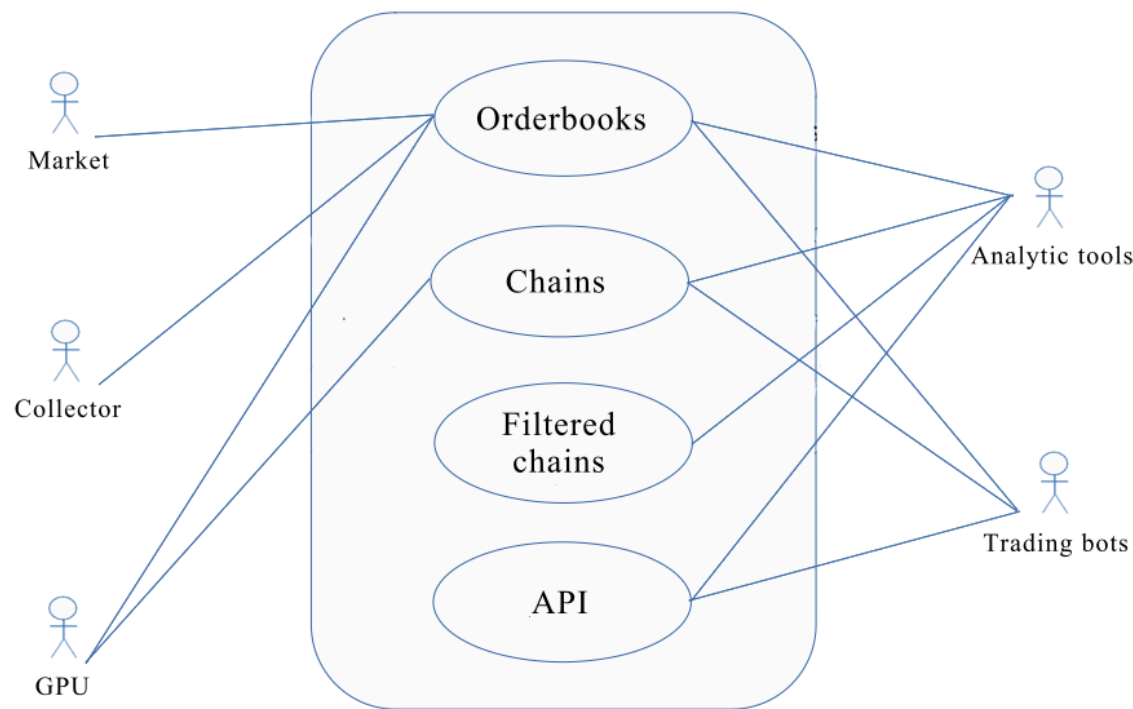
Additionally, the server manager is responsible for generating all possible variations of arbitrage chains.

4. HETHA.IO. ARCHITECTURE

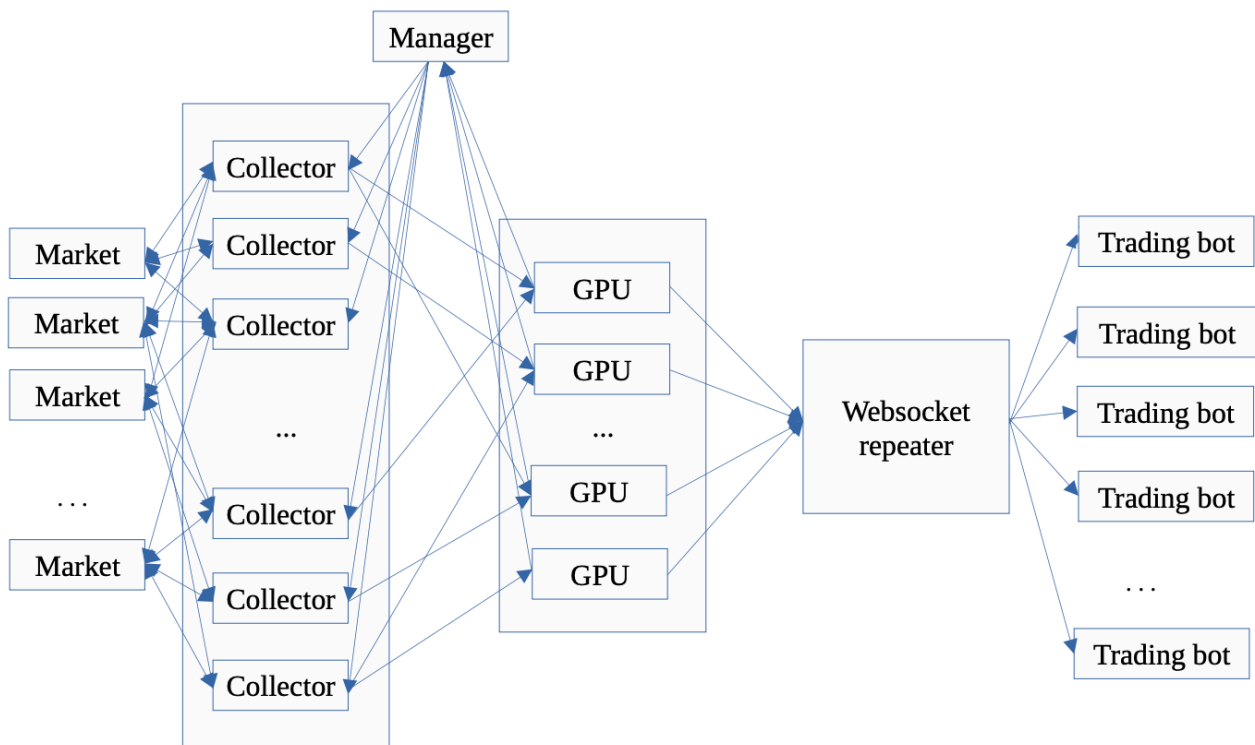
Activity Diagram



Use case diagram



Hardware Diagram



5. RISKS VS PROFITABILITY. HOW TO INCREASE THE PROFITABILITY OF AN ARBITRAGE DEAL. TRENDING, DISCOUNTS.

Filtering chains is not the only method to increase profitability. There are at least two other methods: trending and discount. While the first method increases trading risks, the second keeps them at the same level but requires significant funds.

Discounts

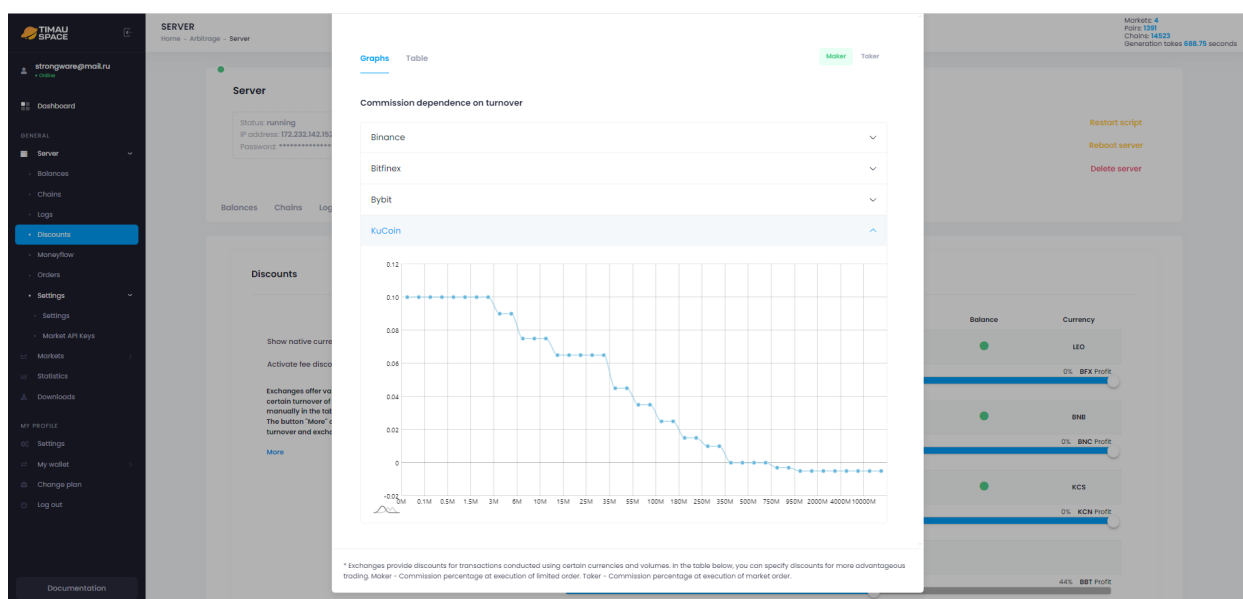
Cryptocurrency exchanges typically offer two types of discounts, which can also be combined in various ways. The combination options are quite individual, so we will focus only on the two main ones:

- Volume discount
- Token discount

To understand discount combinations, it is advisable to visit the website of the platform of interest.

Volume Discount

Cryptocurrency exchanges are highly interested in large players—market makers. Any exchange thrives on trading transactions, and market makers provide liquidity for them. Therefore, trading platforms offer significant discounts on trading fees based on a client's trading volume.



The graph visually illustrates the change in commission percentage depending on trading volumes. The graph shows that when the trading volume exceeds \$500 million per month, the commission percentage on the KuCoin exchange not only drops to zero but even becomes negative.

This is a striking example of the competitive battle for market makers in the industry. However, since arbitrage trading generally involves acting as both a taker and a maker in equal proportions (makers are traders who place orders, such as limit orders, while takers are traders who execute existing orders, such as market orders), one should not rely too much on negative fees. Nevertheless, the reduction in commission percentage as trading volume increases is quite significant.

Example Analysis

Let's analyze the following example: assets totaling \$1,000,000 USDT are distributed across three exchanges.

Total: \$1,000,000 USDT across 3 exchanges = \$320,000 USD per exchange. Exchanges: Binance, KuCoin, Bitfinex. Each exchange holds 6 assets: USDT – \$100,000 USDT, BTC – \$100,000 USDT, SOL – \$40,000 USDT, SUI – \$40,000 USDT, DOT – \$40,000 USDT, ETH – \$40,000 USDT.

Thus, we have created a balanced portfolio (further details on portfolio distribution will be discussed in the following chapters).

In an active market, 3 to 8 trading chains occur per day. On average, we assume 5 trading chains per day. Depth distribution is assumed 50/50 when exchange participation in chains is evenly distributed.

Calculating Trading Volume

$5 \text{ (chains per day)} \times 2.5 \text{ (average chain length)} \times 20,000 \text{ USD (average order size)} \times 30 \text{ (days per month)} \div 3 \text{ (exchanges)} = 2,500,000 \text{ USD monthly trading volume.}$

Now, we check the fee tables on the exchanges that correspond to this trading volume.

	Maker	Taker	Maker	Taker
	2 500 000 USD		Base	
Binance	0.09%	0.10%	0.10%	0.10%
Bitfinex	0.06%	0.20%	0.10%	0.20%
Kucoin	0.09%	0.10%	0.10%	0.10%

From the table, it is evident that Maker fees have received discounts. Now, it is important to understand which specific fees are applied in arbitrage trading.

Fees are determined at the moment an order is placed. In most cases, a limit order is used because it is the only order type that guarantees the required execution price. As mentioned earlier, market orders are classified as taker, while limit orders are typically maker. However, this is not always the case. The probability of a limit order acting as either a maker or a taker is approximately 50%.

Example 1: A client places a limit buy order for BTC/USD at \$69,000. At the moment, the current buy price for BTC is \$70,000. In this case, the order is added to the order book and waits until the price drops to \$69,000, at which point it gets executed. This makes the order a maker.

A similar scenario happens for a sell order. For example, a trader places a sell order at \$71,000, while the current price of BTC is \$70,000. The order is placed in the order book and waits until the price rises to \$71,000 before execution. In this case, the order is also considered maker. Both of these cases classify as maker orders because they add liquidity to the order book rather than taking liquidity from it.

Example 2: A trader places a limit buy order for BTC/USD at \$70,000. However, before the order is placed, the market price drops slightly. Other traders have already placed sell orders at \$69,995. In this case, the order executes immediately, taking liquidity from the order book. Since it removes existing liquidity, it is classified as a taker order.

It is important to note that the order will be executed at a slightly better price than intended. Even though the trader originally placed a buy order at \$70,000, it will be executed at \$69,995 (or lower), depending on the available sell orders in the book.

Since arbitrage trading balances buy and sell orders, there is an equal chance that an order will be executed as either a maker or a taker. As a result, approximately 50% of orders will qualify for a maker fee discount.

Token Discount

This discount is not available on all exchanges, but when possible, it should be used to increase profitability. Some exchanges, including Binance, Bitfinex, and KuCoin, issue their own native tokens. To enhance liquidity and encourage their use, exchanges have developed mechanisms for token utilization. The two primary models for token discounts are using native tokens to pay for trading fees and holding native tokens in the account balance.

For example, on the Bitfinex exchange, holding even a small amount of the native exchange tokens (LEO) grants a 15% discount on Taker fees. If a trader holds 5,000 LEO tokens, the discount increases to 25%, and the tokens are not spent in the process.

On Binance and KuCoin, a different model is used. Their native exchange tokens serve as a means of paying trading fees, so traders must carefully monitor their token balance. Additionally on Binance, using BNB to pay for fees provides a 25% discount on both Taker and Maker fees. On KuCoin, using KCS grants a 20% discount on trading fees.

It is important to note that volume-based discounts and token discounts can be combined for additional savings.

Before starting trading, it is highly recommended to carefully review the documentation of the specific exchange to understand the details of its discount programs.

Below is a breakdown of combined discount programs, based on the previous example, taking into account that discounts apply both for trading volume and native tokens.

	Maker	Taker
Binance	0.065%	0.075%
Bitfinex	0.06%	0.17%
Kucoin	0.72%	0.08%

Let's examine how discounts impact the final profit when calculating an arbitrage chain. Consider the following example:

bnc:SOL/BTC - bfx:SOL/USD - bfx:BTC/USD

The estimated profit for this chain is 0.22%. Chains are always generated based on standard Taker fees, meaning that at each step, a percentage is lost due to fees:

- Binance – 0.1%
- Bitfinex – 0.2%
- Bitfinex – 0.2%

With discounts, the fee losses at each step are reduced:

- Binance – 0.025%
- Bitfinex – 0.030%
- Bitfinex – 0.030%

As a result, the estimated profit increases by: $0.025\% + 0.03\% + 0.03\% = 0.085\%$, leading to a total profit of 0.305%.

As mentioned earlier, the ratio of Maker to Taker trades statistically tends to 50/50. This means that actual fee discounts will be even greater:

- Binance – $(0.035\% + 0.025\%) / 2 = 0.03\%$
- Bitfinex – $(0.14\% + 0.03\%) / 2 = 0.085\%$
- Bitfinex – $(0.14\% + 0.03\%) / 2 = 0.085\%$

The total effective fee in the chain is: $0.03\% + 0.085\% + 0.085\% + 0.22\% = 0.42\%$.

This means that the final profit is nearly twice as high as the initial estimate. Unfortunately, in the HETHA.IO service interface, all displayed profits are nominal. At the moment, actual profit is not calculated.

Trending

Arbitrage is an excellent way to profit in a market moving sideways (Flat). Such market conditions help ensure that all legs of an arbitrage trade are successfully closed. However, markets often move within a range, even if they follow a predictable trend. Although the cryptocurrency market is generally considered unpredictable, it exhibits certain recurring patterns that make trend analysis easier. These include events like halving, the breaking of all-time highs, and other significant milestones. In such cases, trending trading is highly recommended.

In this approach, orders in an arbitrage trade are placed not only according to the arbitrage chain but also with trend consideration. For example, in trend-based trades, an order aligned with the trend can be placed slightly away from the recommended value—meaning it is adjusted in favor of the trend.

Additionally, orders placed against the trend can be set to execute at market price. While this may cause slight price deviations from the initial arbitrage chain, these fluctuations should be compensated by trend-based adjustments. Essentially, the entire arbitrage chain is shifted in the direction of the trend.

When performing arbitrage trading in a trending market, it is crucial to avoid leaving unclosed legs against the trend.

Example of Trend-Based Arbitrage Trading:

We choose a Long trend for the base currency USD, force an order against the trend, and set a 2% shift parameter.

Some traders worry that executing orders at market price introduces significant slippage and price deviations. While there is a chance of this happening, price movements are random at any given moment. This means that with a large enough sample size, the price distribution will be uniform around the expected value.

Empirical evidence supports this: the deviation in price variations remains evenly distributed around the expected value.

The expected value is the price level calculated in the arbitrage chain, used as the reference point for placing limit orders. Based on this, a 0.5% additional yield is generally enough to offset market order execution costs.

However, for higher profitability, the order shift can be increased to 1% or even 2%. The greater the deviation, the harder it becomes to close arbitrage legs, increasing risks but also boosting returns.

Returning to the Example:

We have an arbitrage chain with an initial profit of 0.08%:

$$\text{bbt:SOL/USD} \rightarrow \text{bnc:SOL/USD}$$

Prices in the chain:

bbt:SOL/USD – 190.95 USDT, Buy order

bnc:SOL/USD – 191.50 USDT, Sell order

[illegible]

A Buy order is considered an order against the trend, since the trend is Long. This means that crypto asset prices are in an uptrend, so the asset should be bought as quickly as possible.

To achieve this, the bbt:SOL/USD order should be executed as a market buy.

Since this is a live order that has actually been executed, we can check the Bybit user dashboard to verify the exact trade price.

According to Bybit's data, the actual trade was executed at 190.82 USDT.

BYBIT

Exchange

WEEK

Buy Crypto

Markets

Trade

Tools

Finance

Card

Learn

Rewards Hub

More

USDC 10/13/2024

Deposit

Assets

Orders

Spot

Unified Trading Account

Spot Orders

USDT Perpetual

Inverse Perpetual

Inverse Futures

USDC Perpetual

USDC Futures

USDC Options

Spot (Pre-UTA)

USDT Perpetuals (Pre-UTA)

Inverse Perpetuals (Pre-UTA)

Inverse Futures (Pre-UTA)

USDC Perpetuals (Pre-UTA)

USDC Options (Pre-UTA)

Tools History

Buy Crypto

Earn

Pre-upgrade Order History

Current Orders

Order History

Trade History

Subscription & Redemption

Spot Pairs

All Assets

/

Quote Assets

Trade Type

All

Order Type

All

Direction

All

Order Time

2024-10-31

-->

2024-11-07

Reset

Export

Spot Pairs	Trade Type	Order Type	Direction	Filled Value	Filled Price	Filled Qty	Trading fee	Filled Time	Transaction ID	Order ID
SOL/USDT	Spot	Market	Buy	106.83836 USDT	190.82 USDT	0.508 SOL	0.001608 SOL	2024-11-07 16:41:40	90678979	7740864
SOL/USDT	Spot	Market	Buy	400.72200 USDT	190.82 USDT	2.100 SOL	0.0021 SOL	2024-11-07 16:41:40	9067818	7740864
SOL/USDT	Spot	Limit	Sell	663.00440 USDT	164.11 USDT	4.040 SOL	0.6630044 USDT	2024-11-05 13:50:02	88402070	85729024
ADA/USDT	Spot	Limit	Sell	65.620680 USDT	0.3230 USDT	203.16 ADA	0.06562068 USDT	2024-11-04 23:02:22	66135022	66479316
ADA/USDT	Spot	Limit	Sell	134.749140 USDT	0.3230 USDT	417.18 ADA	0.13474914 USDT	2024-11-04 23:02:22	66135021	66479316
ADA/USDT	Spot	Limit	Buy	800.000343 USDT	0.3217 USDT	2,486.79 ADA	2.48679 ADA	2024-11-04 22:41:17	66071311	46414336
SOL/USDT	Spot	Limit	Buy	379.74996 USDT	169.38 USDT	2.242 SOL	0.002242 SOL	2024-10-31 15:47:05	85553377	46206465

* Only the records of the last 6 months after the account upgrade are displayed. To view pre-upgrade orders, click Pre-upgrade Order History Details.

Thus, we instantly closed one leg using a buy market order at a price better than the one calculated in the arbitrage chain.

The second Sell order was placed at 191.50 USD plus a 2% price shift in the direction of the trend. As a result, the order was placed at 195.27 USD.

Buy Crypto
Markets
Trade ▾
Futures ▾
Earn
Square ▾
More ▾

Deposit

Dashboard
Open Orders
Order History
Trade History

Assets ▾

2024-10-31 → 2024-11-07

Base All ▾ Quote All ▾ Side All ▾

Sort By Order Time ▾ Search Reset

Hide all cancelled

Orders ▴

Spot Order

P2P Order

Transaction History

Earn History

Fiat Order

Convert History

Payment History

Rewards Hub

Referral

Account ▾

Sub Accounts

Settings

Order Time	Pair	Type	Side	Average	Price	Executed	Amount	Amount per Iceberg Order	Total	Trigger Conditions	TPLG	Status
▲ 2024-11-07 16:41:39	SOL/USDT ▲	Limit	Sell	195.27	195.27	3.705	3.705	0	723.47535 USDT	-	-	Filled
<div>Order No. 8233516961 ⓘ Time Updated: 2024-11-07 18:49:21 Total Transaction Fee: 0.00090432 BNB</div> <div> <div>Date</div> <div>Trading price</div> <div>Executed</div> <div>Transaction Fee</div> <div>Total</div> <div>Role</div> </div> <div> <div>2024-11-07 18:49:21</div> <div>195.27</div> <div>3.705</div> <div>0.00090432 BNB</div> <div>723.47535000 USDT</div> <div>Maker</div> </div>												
▼ 2024-11-07 12:53:40	ETH/USDT ▲	Market	Buy	2803.25	Market	0.1284	0.1284	0	359.1373 USDT	-	-	Filled
▼ 2024-11-05 13:41:33	BTC/USDT ▲	Limit	Buy	68907.47632675	68923.11	0.00961	0.00961	0	662.2008475 USDT	-	-	Filled
▼ 2024-11-05 13:41:32	SOL/BTC ▲	Limit	Buy	0.0023722	0.0023722	4.043	4.043	0	0.0059079 BTC	-	-	Filled
▼ 2024-11-04 23:02:22	ADA/BTC ▲	Limit	Buy	0.0000048	0.0000048	620.8	620.8	0	0.00297984 BTC	-	-	Filled
▼ 2024-11-04 22:37:02	BTC/USDT ▲	Limit	Buy	67000.01	67062.57	0.0123	0.0123	0	824.100123 USDT	-	-	Filled
▼ 2024-11-04 22:37:02	SOL/USDT ▲	Limit	Sell	157.28	157.28	5.25	5.25	0	825.72 USDT	-	-	Filled
▼ 2024-11-04 22:37:02	SOL/BTC ▲	Limit	Buy	0.0023335	0.0023363	5.254	5.254	0	0.0122602 BTC	-	-	Filled
▼ 2024-11-04 22:29:50	ADA/USDT ▲	Limit	Sell	0.3253	0.3252	1397.8	1397.8	0	454.70434 USDT	-	-	Filled
▼ 2024-11-04 21:29:56	BTC/USDT ▲	Limit	Buy	67642.88	67642.88	0.00614	0.00614	0	415.3272832 USDT	-	-	Filled
▼ 2024-11-04 21:29:56	AVAX/USDT ▲	Limit	Sell	23.15	23.15	17.96	17.96	0	415.774 USDT	-	-	Filled

In the image above, you can see that the order placed with a 2% shift was closed within 2 hours, resulting in a final profit of 2.08% or, in monetary terms, a profit of 15.04 USD.

It is important not to overuse trend-based shifts when trading in trending mode. The goal of shifting is to compensate for market orders placed against the trend, so the optimal shift size is 0.3% – 0.5%.

6. PORTFOLIO. HOW TO EVALUATE REAL PROFITABILITY

In the course of trading, a reasonable question may arise: what is the actual profitability of the trading portfolio? Due to the diversity of assets, the dynamic value of the portfolio, and the uncertainty of the funds used, determining profitability is quite a complex task.

Therefore, HETHA.IO offers a standardized model for evaluating portfolio profitability — it is the calculation of profitability based on basic arbitrage trades. The essence of the method is to calculate the minimum possible profit from the trades.

However, this method has a significant drawback — low accuracy of calculations. This method calculates the profitability of the arbitrage trade based solely on its nominal profit and the multiplier used during the execution of the chain.

There are many factors that can significantly increase profitability:

- Discounts. Discounts are not considered when calculating profit. They can be used both to increase profit and to simplify the process of closing legs of the chain.
- Maker/Taker fees. These fees can substantially increase the profitability of arbitrage trades. For example, on Bitfinex, the difference between these fees is 0.1%. If Bitfinex appears twice in the chain, the difference becomes 0.2%. In this case, the total profitability of the arbitrage chain might have been 0.05%. Thus, the final profitability increased five times, reaching 0.25%.
- Trend profitability. This factor can drastically change the profit level, for instance, from 0.05% to 2.05%.

The impact of these factors on profitability is not reflected in the final portfolio profitability. This is because the APIs of different exchanges vary significantly, and it is not always possible to clearly determine the size of the commissions. As a result, the profit currently visible in the user's dashboard is nominal, not actual. The actual profit will definitely be higher.

Having estimated the approximate minimum profit from the arbitrage chains, the next step is to evaluate the profit percentage.

For example, by reviewing the statistics for the past month, it might be shown that the profit was 825.00 USD. Given that the portfolio, according to balance data, amounts to 524,000 USD, the calculation is:

$$825.00 \text{ USD} / 527,000 \text{ USD}$$

This results in a 0.1565% monthly profitability.

Is this approach correct? No. The income should be evaluated not relative to the portfolio size, but relative to the volume of funds involved in trading.

How to Determine the Used Funds?

To determine the used funds, go to the Moneyflow section. In this section, you can view variables that characterize the monthly trading data: the number of chains, the number of orders, the turnover, and the funds used. The last parameter is the one that reflects the funds involved in generating profit.

Below is an example of calculating the funds involved from the portfolio. The portfolio is the source of funds for turnover. If the funds from the portfolio have already been involved in the turnover and are sufficient for the current trades, no additional funds are allocated from the portfolio.

BNC		BFX		Итого в обороте
From portfolio	On balance	From portfolio	On balance	
bnc:btc/USD-bfx:btc/USD – 100 USD				200 USD
USD: -100 USD	BTC: +100 USD	BTC: -100 USD	USD: +100 USD	
bfx:btc/USD-bnc:btc/USD – 50 USD				100 USD
USD: +50	BTC: +50 USD	BTC: +50	USD: +50 USD	
bnc:btc/USD-bfx:btc/USD – 100 USD				300 USD
USD: -100 USD	BTC: +150 USD	BTC: -100 USD	USD: +150 USD	
bfx:btc/USD-bnc:btc/USD – 50 USD				200 USD
USD: +50	BTC: +100 USD	BTC: +50	USD: +100 USD	

The amount of funds used is the maximum value over the period. In this case, 300 USD.

Returning to the example: number of chains: 115, number of orders: 273, turnover: 1,734,636 USD, used assets: 239,205 USD. Thus, the profitability of the arbitrage chains is 0.3449% per month. However, it should be noted that the profit of 825 USD is nominal. Therefore, all chains need to be recalculated using the exchange's client dashboards with accurate information for all orders. After recalculating all the completed chains considering the discounts, it can be concluded that the final profit was 1,503 USD. Thus, the final profitability is $1,503 \text{ USD} / 239,205 \text{ USD} = 0.63\%$ per month, or 7.54% per year.

7. PORTFOLIO. CREATING A PORTFOLIO

Creating an investment portfolio is one of the most important tasks. It should be noted that there are two different types of arbitrage: conservative (classical) and advanced.

The conservative type of arbitrage involves having your own assets without any leverage. The assets must be placed on the corresponding exchange and in the appropriate currencies. Depending on the allocation of funds, different arbitrage deals will be executed.

When creating an investment portfolio for a conservative model, it is strongly recommended to follow these rules:

- Use only liquid pairs and liquid exchanges;
- Use exchanges with well-defined rules for deposits and withdrawals;
- Do not use pairs subject to Pump / Dump tactics;
- Do not use unknown or dubious tokens;
- A significant part of the portfolio should be placed in base currencies;
- USDT in the portfolio.

Use only liquid pairs and liquid exchanges

Liquidity is the main condition for arbitrage. Trading in pairs where no transactions take place or very small volumes are used is highly risky. This greatly increases the risks of unclosed legs of arbitrage chains.

Use exchanges with well-defined rules for deposits and withdrawals

It is crucial to check the possibility of transferring funds between exchanges. This primarily concerns fiat currencies. Generally, fiat currencies should be avoided, as processing fiat funds through bank accounts is expensive not only from a financial standpoint but also from an AML (Anti-Money Laundering) perspective. However, fiat currencies are not the only type of problematic assets.

It is also necessary to verify the possibility of depositing and withdrawing each currency. Additionally, it is important to check the compatibility of token transfer networks. Typically, besides major cryptocurrencies, most of them are tokens that work on different blockchains. It is important to check if the input and output of specific tokens correspond to the same blockchain. Otherwise, transfers between exchanges will be impossible.

An example that was mentioned earlier: WRX constantly generates arbitrage chains. However, it has restrictions on deposits and withdrawals on the Kucoin exchange. In this case, WRX should be removed from the allowed chains.

Another example: JST can only be withdrawn from the exchange via the BSC network (TRC20 is temporarily disabled), while deposits of this currency on Kucoin and Bybit in BSC are not possible (only TRC20).

All options for migrating funds should be studied before including a particular currency in the arbitrage portfolio.

Do not use pairs exposed to Pump/Dump schemes

Let's break down the nature of such schemes. There is a cheap cryptocurrency with a small market capitalization. The organizers of the Pump/Dump operation actively buy it to push the price up. Trading on the market increases, and regular traders join in to support the Pump. Once certain price levels are reached, the organizers dump all their holdings at the inflated price. After that, the price of the cryptocurrency sharply drops.

The area of risk is particularly significant for low-market-cap assets, as their prices are easier to manipulate. You should be very cautious when including such assets in your portfolio. Pump/Dump operations are very short-lived – the trading at the highs can last only a few days. Furthermore, the price of a cryptocurrency can spike very quickly to its maximum. In such cases, the risk of having unclosed legs is very high. Additionally, the asset's price can fall to zero, devaluing part of the portfolio. Therefore, these currencies should be handled with great caution.

Do not use unknown or questionable tokens

This point is essentially similar to the previous one. However, it should be noted that sometimes you can find interesting and functional tokens, despite their low liquidity.

You should carefully study their idea, business model, team, and plans. From an investment perspective, it sometimes makes sense to buy new, unknown tokens. For example, placing a small part of the portfolio in SUI at the time of its exchange launch was a reasonable decision. However, this is more of an exception than the rule. The overwhelming majority of crypto projects are not particularly interesting.

A significant portion of the portfolio should be allocated to base currencies

Base currencies are the primary currencies represented in the market. As a criterion for defining a base currency, you can refer to the coins listed in the TOP 15 on the site: <https://coinmarketcap.com/coins/>. However, even within this top, there are exceptions, such as meme coins.

A promising cryptocurrency should possess three factors — liquidity, competitive functionality, and infrastructure. A significant portion of the portfolio, about 85-95%, should consist of base currencies. The remaining 5-15% can be used for riskier experiments.

It is important to remember that cryptocurrencies have no guaranteed underlying assets. Moreover, there is no consistent audience support. Liquidity can easily flow from one crypto asset to another as investor sentiment changes. Given that large players like BlackRock have entered the crypto market, the risks of holding crypto assets are gradually being mitigated. However, investors cannot fully shield the crypto market from regulatory actions.

You should adhere to the rule — the more liquid the pair, the lower the volatility, and therefore, the lower the risks of catastrophic price declines for crypto assets.

USDT in the portfolio

Experience in trading, particularly in arbitrage operations, suggests that trading in USDT is generally more active than in BTC. Therefore, the share of USDT in the portfolio should be greater than that of BTC. Moreover, USDT can be considered a sort of safety net during market crashes, allowing you to buy assets at a lower price.

To summarize the above points, when constructing the portfolio, the following rules should be considered:

- More USDT
- A little less BTC
- A few assets from the TOP 15
- In total, USDT + BTC + TOP 15 < 85-95%
- Interesting assets 5-15%

Let's create an example portfolio of 50,000 USD:

Coin	%	USD
USDT	35	17 500
BTC	25	12 000
SOL	15	7 500
ETH	15	7 500
SUI	5	2 500
TON	5	2 500

A portfolio with a large share of USDT will have a high degree of inertia. This means that it will grow and fall in a relatively controlled manner. Moreover, such a portfolio will allow for active earnings from cryptocurrency arbitrage.

For working in the advanced arbitrage model, a portfolio as such is not necessary, since this type of arbitrage uses only a balance in USDT. On the one hand, this is convenient, as cryptocurrencies, unlike stablecoins, are subject to very high volatility. However, a balance consisting only of USDT will not grow along with the market. Yes, in this case, it certainly won't fall, but it also won't grow.

The main task is to create a portfolio that can grow effectively due to market conditions and, in addition to that, passively increase through arbitrage deals.

8. PORTFOLIO. HETHA.IO ANALYSER – FINDING THE RIGHT PAIR

HETHA.IO Analyser allows you to analyze arbitrage chains. This tool works exclusively with filtered chains.

It is impossible to analyze the flow of unfiltered chains, as it is too large, and currently, it is not feasible to store such data. In the user cabinet, unfiltered chain data is available only for the last hour, after which it is discarded. However, each user can create their own mechanism for storing unfiltered chains in a database for further analysis.

Additionally, each user can, based on already available scripts, set up their own filter and create their own database of filtered chains that meet their personal requirements. The HETHA.IO Analyser, in turn, works with the built-in filter.

There are several important points to consider when searching for the right pair:

- Search without reverse chains;
- Search with reverse chains;
- Search by portfolio;
- Search by income.

Although the analyzer allows searches for no more than 7 days, it is strongly recommended to perform searches for the pair of interest over a period of 2-3 months in several iterations. This is necessary to form a full picture of the emergence of arbitrage events. One should not rely on search results for just one week, even despite the abundance of chains and large profits. Systematic deviation can only be detected over an extended period.

Search without reverse chains

The issue of using reverse chains has already been discussed. We previously talked about the benefits of using them, and now we will discuss situations where they can be disregarded.

The possibility of ignoring reverse chains is determined by the fee for transferring funds. The most expensive for transfers are BTC and ETH. BTC is the absolute leader in terms of transfer costs. At the same time, many tokens use cheap networks for transfers, such as Optimism, Polygon, TRC20, SOL, TON. This allows balancing separated balances with minimal expenses, and sometimes even without any costs.

For example, for the USDT token, it is almost always possible to find a way for free transfers from exchange to exchange. For instance:

- Bitfinex transfers USDT for free across all networks except ERC20.
- Binance transfers USDT for free in the BSC network.
- Bybit transfers USDT for free in the Mantle Network.
- KuCoin transfers USDT for free in the BSC network.

There are many possibilities for transferring any amount in USDT for a fee of 0.5 cents or less. Therefore, when working with USDT tokens, reverse transactions are not necessary.

Additionally, reverse transactions are not needed in advanced margin trading methods since no actual asset is acquired in this case. All trading is done with borrowed assets, and the focus is solely on the profit.

Search with Reverse Chains

BTC transfers can cost between 10 - 20 USD, and ERC20 transfers 5 - 15 USD depending on the network load. This is quite expensive, so it is necessary to search for chains with reverse transactions.

When searching for reverse chains, preference should be given to more frequent occurrences of direct and reverse chains rather than their average profit. The systematics of events are important, not isolated spikes. These spikes could be symptoms of non-systematic deviations, such as a PUMP, which may not repeat.

Search by Portfolio

If the portfolio is already formed, the analyzer can help find a more profitable allocation of funds. For example, a portfolio contains SOL worth 6,000 USD. Theoretically, without the analyzer, the funds should be distributed across 3 exchanges where you plan to trade: Binance, Bybit, and Bitfinex. However, the analyzer shows that arbitrage chains occur systematically on Binance and Bitfinex, but rarely with this combination on Bybit. It makes sense to distribute the funds not across 3 exchanges with 2,000 USD each, but across 2 exchanges with 3,000 USD each. Increasing the deposit by 1.5 times will proportionally increase the profitability of such arbitrage deals.

Search by Profit

This method is suitable for arbitrage deals conducted with the advanced method. In this case, only the profit matters. Neither the frequency of transactions nor reverse transactions affect trading since no real asset is acquired; only USDT is used. What matters here is only the profit and the coin's name.

For the coin's name, an analysis should be done as mentioned earlier: liquidity, migration freedom, PUMP/DUMP probability. If after analysis everything looks convincing, then the coin should be added to the list of allowed coins for receiving chains and order books with it, thus enabling the script to work with it.

The list of such coins should be quite large. This is necessary to provide the script with trading options. Therefore, time should be spent on this analysis.

A more detailed look at the analyzer interface is covered in sections 15-18, which describe the product interface and the process of working with it.

9. TRADING BOT. SETTINGS OVERVIEW

When you get a product, you also get a set of default settings, but you can change them if you want. To do this, go to the server, go to the/home/trader/libs folder. There you will find the file called params.json. To change the settings, you must open the file and change its values to the desired.

Below are all these parameters with an explanation of the values:

ROOT Module

- "release": 2.4.5,
 - Version.
- "mode pro": true,
 - Version type (Pro or Standart).
- "debug": false,
 - Enable/disable system messages.
- "lock_timeout": 3,
 - Timeout when new chains using the same currency as the previous one will not start. The time is in milliseconds, it means that by default it is set 3 minutes from the last deal in the last chain to start a new deal in the new chain using the same currency as a previous one.
- "percent_balance": 0.9,
 - The deal amount must not exceed the specified value of funds on the account. By default it is set 90%.
- "gentime limit": 1,
 - Max. allowable chain generation time, sec.
- "profit limit": 110,
 - Display in the logs the chains with profitability from VALUE, %.
- "chainlog limit": 1000,
 - Display VALUE rows in chains.log.
- "ws interval": 1,
 - Reconnect to the server in VALUE, min.
- "balance interval": 2,
 - Balance information is requested every VALUE, min.
- "allow get balance": true,

- Enable/disable displaying balances in the personal account.
"allow update": false,
- Enable/disable "Update" button.
"callback port": 3003,
- Callback port.
"ws url": ws://feed.hetha.io:8080/ws/.
- Server address.

WS PORT Module

- "pro": 8080,
- Pro version port.
"standard": 8000,
- Standard version port.

WS PARAMS Module

- "event": cups,chains,
- Event - If "chain" is specified, all chains with profitability greater than 100% will be transmitted. If the minimum profitability percentage is higher, it is specified as "chain_100.05". "cups" - all cups are transmitted.
"market": bnc knc bbt bfx,
- Receive data from VALUE exchanges.
"currency": btc usd ada avax eth sol sui trx,
- Receive chains with VALUE currencies.
"profit": 100.02,
- Receive chains with profitability above VALUE, %.

CHAIN BASE Module

- "0,1,2": btc usd
- Base currency. If the chain doesn't include the base currency, it won't be processed.

STORAGE

- "balance": true,
- Permit or prohibit the retention of balance information.

"balance backup hour": 18,

- Balance information is saved every day at VALUE, h.

"chain": true,

- Permit or prohibit the retention of chains.

"order": true,

- Enable/disable order saving.

AI Module

"accept": true,

- Enable/disable AI usage. If this setting is enabled, order data will be available to the service for model training.

"pro enable": true,

- Enable/disable forecasting of chain execution time.

"pro threshold": 0,

- Threshold value for chain execution time. Below this value - allow execution, above this value - prohibit execution, min.

"base enable": true,

- Enable/disable forecasting of chain execution.

"base threshold": 0,

- Threshold value. Chains are not executed with a coefficient higher than the VALUE.

FORCE REVERSE Module

"accept": true,

- Enable/disable balance distribution mode.

"chain length": 2,

- Max. allowable chain length.

"percent_accept": 0,

- Min. allowable profit for the reverse deal.

"activator": 4,

- Force reverse is activated if the number of base dimensions (\$100 or equivalent) is less than

VALUE.

"multiplier": 8,

- Increase the sum of the chain by VALUE times relative to the base dimension (\$100 or equivalent).

LITE Module

"accept": true,

- If the previous conditions were executed or were not executed, the chain is either accepted or not accepted.

"percent_accept": 0.2,

- Minimum profit is not less than the specified value. By default it is set 0.2%. "force_trading": 3,
- Use "Market" order placement type if profit exceeds set value (%). "rate_diff": 0.05,
- The chain is accepted if the deviation from the initial rate is not more than the specified value (%)

"multiplier": 2

TREND Module

"accept": false,

- Accept Trend deals.

"usd": long,

- Long - Buy at the current price (limited order) and sell when the price increases by «Offset by VALUE, %». Short - Sell at the current price (limited order) and buy when the price decreases by «Offset by VALUE, %». Flat - Sell at price « $\frac{1}{2}$ *Offset by VALUE, %» higher than current, and buy « $\frac{1}{2}$ *Offset by VALUE, %» lower.

"btc": flat,

- Long - Buy at the current price (limited order) and sell when the price increases by «Offset by VALUE, %». Short - Sell at the current price (limited order), and buy when the price decreases by «Offset by VALUE, %». Flat - Sell at price « $\frac{1}{2}$ *Offset by VALUE, %» higher than current, and buy « $\frac{1}{2}$ *Offset by VALUE, %» lower.

"force_against": true,

- If True is selected, then:

Long – Buy at the current price (market order) and sell when the price increases by «Offset by VALUE, %».

Short - Sell at the current price (market order), and buy when the price decreases by «Offset by VALUE, %».

"offset by": 0.2,

- Sale or purchase when the price changes by VALUE, %.

FULL Module

- "accept": true,
- If the previous conditions were executed or were not executed, the chain is either accepted or not accepted.
"percent_accept": 0.1,
- Minimum profit is not less than the specified value. By default it is set 0.1%. "force_trading": 1.5,
- Use "Market" order placement type if profit exceeds set value (%). "rate_diff": 0.1,
- The chain is accepted if the deviation from the initial rate is not more than the specified value (%)
"multiplier": 4
- Multiplies chain sum on a specified number of times.

10. TRADING BOT. MAKING EVERYTHING SAFE

The trading bot has a variety of settings that can be accessed through the admin panel. However, it is important to note that managing the settings of the trading bot in the admin panel is convenient but not overly secure. This is especially true when it comes to exchange keys. However, there are several actions that can make the storage of access keys to exchanges relatively safe.

First, let's discuss how the server creation works. To create a dedicated client server, HETHA.IO uses the platform linode.com. With its help, a simple server is created from a pre-prepared image. This image is created within our company. It has the Centos Stream 8 operating system installed, node.js with the required libraries, and the product itself. Nothing else is pre-installed on

the server. No backdoors, keyloggers, or any other spyware. The firewall has several ports open strictly within the required scope.

All of this can be verified by gaining access to such a server. When creating the server, it is necessary to specify a password for the root user. The service randomly generates and provides it during the creation process. After the server is created, the user can find its IP address and root password in their account.

The server can be fully managed from the user's account, but one can also use an SSH client to directly access the server. Moreover, the specified root password can be changed to another one to exclude the old password from our system. The old password will still be visible in the account, even though it becomes obsolete.

Changing the password:

```
sudo passwd root
```

Creating API exchange keys

Typically, keys consist of two parts: Key and Secret. Sometimes there may be more parts, but the essence remains the same. The keys need to be created and saved. It is important to remember that these keys can give the ability to operate funds on the exchange account not only to the person who created them but also to anyone who gains access to them. Therefore, they must be stored very securely.

However, to reduce the risks in case of a key leak, it is essential to pay attention to the fact that all exchanges offer the ability to set permissions for specific keys. For trading via the HETHA.IO trading bot, it is necessary to minimize the permissions. You should remain:

- Spot trading operations
- Account information

What must definitely be removed:

- Withdrawal of funds from the account

What is recommended to remove:

- Margin/Futures trading
- Moving funds across different accounts
- Any other features that are not related to the operation of the HETHA.IO script.

It is highly recommended to specify the IP address from which trading operations can be conducted. The client's server IP address can be seen in the "Server" section header. However, this feature does not work correctly on all exchanges.

By creating keys in this way, it is quite safe to store trading keys in the HETHA.IO database to avoid configuring the server via terminal. However, if there is a need for an even higher level of security, the configuration settings should be moved to the server.

To do this, you need to go to the user account /Server/Settings/Settings and find the “JSON” button in the Settings section. When clicked, a window will open in which all the parameters set in the account will be displayed in JSON format. These need to be copied, then, using an SSH terminal, access the server, go to the folder /home/trader/libs, and paste the data into the params.json file.

After changing the root password, you can switch the storage of system settings from the shared database to the server. To do this, in the user account settings at /Server/Settings/Market API Keys, enable the "Local key storage" option and then reboot the script using the "Reboot script" button.

A small note. Even though the client is provided with a full Linux server to work with the HETHA.IO script, the service usage rules prohibit clients from installing any additional software on the rented HETHA.IO server that is indirectly unrelated to the HETHA.IO product. This primarily concerns any software related to traffic generation, mail servers, databases, and so on. The server is provided to the client solely as a platform for running the trader scripts. If it is discovered that the client is using the server for purposes other than intended, HETHA.IO reserves the right to terminate the server rental agreement and delete the server.

11. TRADING BOT. MANAGING THE TRADING BOT, FINE-TUNING SETTINGS, AND CHECKING OPERATION

This section will cover the settings of the trading bot. This information might overlap with the section related to the interface; however, in this block, the settings will be discussed as part of the trading strategy rather than as technical parameters. All of the parameters listed below affect the system's profitability:

- *Lock_timeout()*

Default value: 3 minutes. This parameter is responsible for locking chains to prevent them from being used too frequently. Chains typically appear in waves. For example, a chain:

BTCUSD-BTCUSD profit 0.05% - 0 second

BTCUSD-BTCUSD profit 0.10% - 5 second

BTCUSD-BTCUSD profit 0.12% - 25 second

BTCUSD-BTCUSD profit 0.04% - 26 second

BTCUSD-BTCUSD profit 0.01% - 34 second

Part of this wave is filtered at the chain generator side. For example, if the profit change is too small, such chains will be deleted during the generation stage.

BTCUSD-BTCUSD profit 0.1200% - 25 second

BTCUSD-BTCUSD profit 0.1201% - 26 second

BTCUSD-BTCUSD profit 0.1202% - 26 second

Also, chains that are generated at a rate of hundredths of a second will be trimmed to avoid spamming the WebSocket connection.

BTCUSD-BTCUSD profit 0.1200% - 25.00 second

BTCUSD-BTCUSD profit 0.1200% - 25.01 second

BTCUSD-BTCUSD profit 0.1200% - 25.02 second

This chain generation frequency needs to be processed in order to reduce the volume of chains being processed.

- *LockCurrencies()*

This function locks the assets that have just been used during the lock timeout. For example, a chain is launched for execution: bfx:ethusd-bnc:ethusd. From the chain, it's clear that ETH needs to be bought in the first step and sold in the second step. After selling any of the assets, operations with that asset must be paused until the active balance is determined. Thus, BFX:USD and BNC:ETH will be locked for 3 minutes until the new balance state is verified, meaning that all incoming chains using BFX:USD and BNC:ETH will be blocked.

However, chains using BFX:ETH and BNC:USD will not be blocked because during the execution of the chain above, these balances may only be replenished. Balances are updated at the `balance_interval` frequency. It is recommended to set the `lock_timeout` and `balance_interval` parameters in such a way that `balance_interval` is shorter than `lock_timeout`.

- *BlockChain()*

This function locks chains during the `lock_timeout` period if any of the assets are missing. Therefore, balances should not be checked until they are updated. These parameters are undoubtedly interrelated. There is no need to set such long blocking periods. It is quite possible to reduce them to 30 and 20 seconds, respectively. The main limitation is the permissible frequency of balance queries to the exchanges. Different exchanges have various rules regarding the frequency of REST requests. Therefore, before reducing this parameter, it is important to inquire about the market requirements.

Direct Chain Settings (Full / Lite)

- *percent_accept*

This is the profit percentage above which a chain is considered. On the one hand, reducing the percentage when reviewing chains increases the number of chains being processed, but on the other hand, it is important to consider the wave-like nature of incoming chains. Usually, chains do not occur in isolation, and after one chain, there may be an entire sequence, with the most profitable chain typically being neither the first nor the second, but the third or fifth. When starting with a low threshold value, there is a risk of capturing the first, but not the most profitable chain, causing the corresponding assets to be blocked and filtering out the profitable chains.

Additionally, it is important to take into account that any execution of chains and, consequently, the opening of live orders is quite risky. Quickly closing all legs of a chain is likely but not always achievable. Therefore, sometimes chains may remain in an open state for a long period. While the risk is small, a more justifiable arbitrage strategy might be "less often, but larger" rather than "more often, but smaller." The risk in arbitrage is related to the number of chains executed, not their profitability.

- *Force_trading*

Default value: 1. This is a somewhat controversial parameter. It allows executing market orders instead of limit orders for chains with a profit of more than 1%. The advantage of this setting is that the chain will be 100% closed, and all orders will be executed. However, there are also drawbacks:

1. It is unclear at what price the orders will close. For example, a 1% price difference in Bitcoin today could be around \$1000. This seems reliable because even if the prices start to move, the price is unlikely to overcome such a large difference in a few seconds. However, during times of high volatility, even a \$1000 difference can be overcome almost instantly.
2. Additionally, as mentioned earlier, all chains are calculated during generation based on market order prices. When limit orders are placed, market makers often provide discounts, which results in additional profitability. The value of this parameter should be set thoughtfully, as while it eliminates some risks, it introduces others.

- *Rate_diff*

The script receives not only arbitrage chains via Websocket but also order books. These order books are used to recalculate and verify the chains. During the generation of chains, the order books could have changed, which means that a chain may become outdated by the time it reaches the trading

bot. This parameter checks the price deviation between the prices indicated in the chain and the current prices in the trading order book, directly regulating trading risks. Increasing this parameter reduces the number of chains that can be accepted for development.

This parameter can be paired with the `gentime_limit` parameter (default is 1 second). It is possible to soften one parameter with the other. For example, increasing the `gentime_limit` to 5 seconds while increasing the `rate_diff` parameter. This means that if the chain is old but the prices haven't changed, it can still be accepted for development. On the other hand, if the chain was generated 0.1 seconds ago but has price deviations, it is likely that those deviations are due to the new order book not yet being formed.

- *Multiplier*

All chains in the generator are calculated based on a base transaction size of 100 USD. The `Multiplier` parameter defines the size of the trade and acts as a multiplier for the base volume. If set to 0, the transaction is limited to the minimum link in the chain. The volume is automatically calculated based on `percent_balance` (default is 0.9). This means that the transaction will be made at 90% of the volume of the node in the chain with the smallest assets, but no less than one base volume.

The use of this parameter depends on the strategy of other bot settings. If other settings aim to minimize the number of trades, then this parameter should be left for automatic calculation. However, if the number of processed chains may be larger, the `Multiplier` can be used to attempt to catch the arbitrage wave. Let's revisit the example:

BTCUSD-BTCUSD profit 0.05% - 0 seconds

BTCUSD-BTCUSD profit 0.10% - 5 seconds — Multiplier 8

BTCUSD-BTCUSD profit 0.12% - 25 seconds — Multiplier 8

BTCUSD-BTCUSD profit 0.04% - 26 seconds

BTCUSD-BTCUSD profit 0.01% - 34 seconds

- *Force_reverse*

Reverse trades are designed to automatically address balance separation and are highly recommended. You should not neglect reverse trades, as they help redistribute accumulated funds to other accounts with minimal profit. Otherwise, if they are not used, funds often begin to accumulate in the same place. For example, USDT tends to accumulate on BFX. The flow and movement of funds can be tracked in the "Moneyflow" section. Transfers between exchanges are possible, but assets like Ethereum and Bitcoin are very costly to transfer, so it's recommended to avoid direct transfers. On the other hand, USDT can often be transferred with minimal fees.

- *Trend*

This parameter is recommended to be used when there is a clearly defined trend. Generally, such trades significantly increase profitability, but on the other hand, they can no longer be considered arbitrage. Orders may remain open for a long time without closing the legs of the chain. This type of trade cannot be analyzed with AI indicators.

- *AI*

There are two different neural networks that predict the speed of closing arbitrage chains. The Base model is based on analyzing market conditions, connection speeds, load, etc. The Pro model is based on market indicators, 12 indicators, and minute, 15-minute, and hourly charts.

The training was conducted using thousands of orders previously executed on arbitrage chains in HETHA.IO. Moreover, training is ongoing. The neural network provides accurate predictions in calm markets, so it can be effectively used in combination with the `volatility_limit` parameter.

12. TRADING BOT. DEVELOPING ADDITIONS

Writing and Testing New Trading Modules

When writing new trading modules for new markets, the main challenge is testing the core functions of the module. For this purpose, a special utility called `tc.js` is used, located at: `:/homa/trader/libs/modules/tc.js`. It automatically loads all files from the folder in which it resides as trading modules.

It's important to note that all API keys are taken from the regular `params.json` file. Therefore, before testing, you should verify that all file names in the corresponding sections of the JSON are correct and that the necessary keys are present. If all keys are correctly defined according to the file names, the application will load in terminal mode, and testing for the corresponding module functions can be performed.

Possible example queries:

```
m.bnc.getBalances(callback);  
m.bfx.getOrder("adausd", "168826588211", callback);  
m.bbt.limitOrder("btcusd", "buy", 0.1, 50000, callback);  
m.kcn.marketOrder("btcusd", "buy", 0.1, callback);
```

If the API connection parameters are configured correctly, the functions will execute and return corresponding responses from the markets. If all functions work successfully in the test utility,

the model should function properly in the trading script as well. It's worth noting that your own custom trading module cannot be loaded automatically, as modules are loaded centrally. However, you can manually insert it.

The application below provides a list of tasks for creating your own custom module.

Changes to the Trading Bot at Your Own Risk

You are allowed to make any changes to the trading bot, except for the aforementioned malicious code, spam solutions, brute forces, etc. However, remember that the service has an update system in place. There is a risk that after making some private code changes, the system will perform an update on a general basis, and all your changes will be overwritten.

The update doesn't run automatically; it requires the user to click a button in the user panel to trigger it. However, it is possible to click it accidentally. Therefore, it's essential to set the *allow_update* parameter to false before making any code corrections. After that, you can proceed with making the necessary changes to the trading script code.

In theory, the script could load new libraries when started, including directly in NODEJS. This means that NPM may be run, which could reformat or update the libraries. If any libraries were manually added without using NPM, they might be removed.

The Trading Script Does Not Use Forever

The life cycle of the trading script is maintained by CRON. Therefore, to stop the script, you must use the *cron_stop.sh* script. However, understand that the script will stop for no more than a minute, and this is more like a restart. To fully stop the script, you need to modify the file *etc/cron.d/0hourly*. Comment out the line: `* * * * * root /home/trader/cron_start.sh`, and then run the command: `service crond restart`.

To run the script in permanent mode, you must uncomment the above line and restart the CRON service.

13. TRADING BOT. API, FEEDBACK WITH GUI

The HETHA.IO service has its own API. Using this API, it is possible to send trading data to the GUI for display in the user's dashboard. Of course, sending data, such as the status of orders, is not mandatory, as this data may be considered commercial secrets or simply confidential. However, centralized order storage makes the control system more convenient for the user.

Additionally, to train predictive neural networks, the HETHA.IO service may use clients' order and chain data. This is specified in the service's terms of use as well as in GDPR regulations. Apart from training the neural network, this data is not used by the service for anything else. The service is interested in client order information to improve forecasting quality. Therefore, if a client uses the forecasts from HETHA.IO, the service expects the client's orders to be stored on the server.

Below are the API requests to the api.hetha.io server. Authentication on the api.hetha.io server is automatic. The authorization key is provided to the trading bot upon the first connection. Access to the API server is restricted by an IP whitelist, which is also created automatically as user servers are added. The authorization key is stored in the file: /trade/libs/keyword. If the file is absent or empty, the key will be generated again upon the first connection to api.hetha.io

API Requests to HETHA.IO:

- server/credentials – fetches basic settings of the script. No parameters are required.
- server/modules – fetches centralized trading modules with the option to install them if the module is missing.
- server/estimator – loads the centralized Base model of the neural network.
- stream/forecast – requests an evaluation from the Pro neural network based on the analyzed chain. The chain parameters are in base64 format.
- save/order – requests to save an order. Parameters: chainId, orderId, response, market, pair, side. After saving, the order will appear in the Server/Orders section of the user dashboard.
- save/getopenorders – requests the entire list of open orders for the user. No parameters are required. User identification is done via authorization keys.
- save/closeorder – requests to change the status of an order. Parameters: orderId, status, datetime.

IMPORTANT! The save/order, save/getopenorders, and save/close functions should remain operational and, preferably, in their original state as mentioned above. Data from user models is used for training the Pro neural network. To avoid errors in the training, HETHA.IO asks all users to pay special attention to this matter.

- `save/chain` – requests to save a chain. Parameters: `lite`, `chainId`, `history`, `deep`, `profit`, `chain`, `status`, `comment`, `force`, `multiplier`, `forecast`. Chains saved using this request can be found in the Server/Chains section of the user's dashboard. The status field can be true for a completed chain or false for an incomplete one. Incomplete chains with comments can be collected, for example, saving the most profitable chains. Later, these can be searched in the user's dashboard.
- `save/balance` – saves the user's balance state to the database so that it appears in the Server/Balances section of the user dashboard. The balance is saved simultaneously, which helps ensure daily tracking of portfolio changes and makes it easier to analyze the performance of arbitrage chains and the overall portfolio value. Parameters: `market`, `data`.

For more detailed examples of calling requests with parameters, you should refer to the code of the trading bot.

The trading bot is a web server based on Nginx. Its task is to send logs to the user panel. The logs can be accessed via a regular browser: http://ip_address:9888. This will open the `/trader/logs` folder. These same logs are processed and displayed in the Server/Logs section of the user's dashboard.

Requests to the trading bot are made via the port specified in the bot's settings (`callback_port`, by default 3003). For example: http://ip_address:3003/get_version. The response should be a JSON message. Access to the callback can be restricted using the `allow_callback` setting (by default true):

- `/remove_chains` – allows deleting unnecessary chains from the Server/Chains section. This mainly concerns informational chains. The HETHA.IO service does not clean up user chains automatically; it only happens when the user makes the corresponding request in the user panel.
- `/stop_script` – stops the trading bot. Since the bot will restart after a minute due to CRON, this is essentially a restart. The “Restart script” button in the user dashboard triggers this call.
- `/get_balances` – retrieves balance data for quick display in the Dashboard on the user's panel. This request can be blocked by the `allow_get_balance` setting (by default true).
- `/get_version` – used to show the “Update script” button in the user dashboard if the user's current script version is older than the available update.
- `/update_script` – triggers the script update. This can be blocked using the `allow_update` setting (by default false), as mentioned above. The update must be disabled if the user has started making changes to the trading bot code, otherwise, there's a risk of permanently losing all modifications. The update is never run automatically. To perform it, the user must press the “Update script” button. However, it's important to protect yourself from accidentally clicking it.

14. PERSONAL ACCOUNT. BASIC DIFFERENCES IN TARIFF PLANS

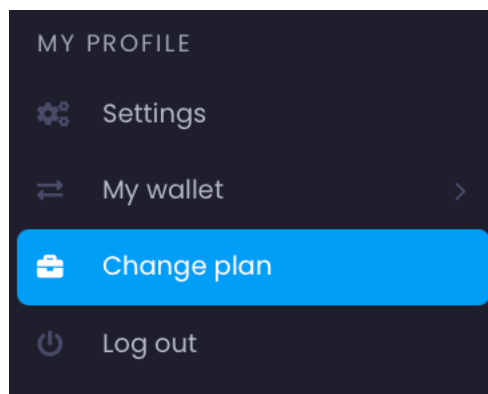
We offer two product variations:

The first option is the **"Free"** plan. It is completely free. With this plan, the client gains access to the "Market" section. This section contains all the analytics for the data types "Stream," "Stream Pro," and "Stream Filtered." Additionally, access to viewing chains and order books is provided. If you are only interested in analytical and statistical data, this option is for you.

The second option is the **" Pro "** plan. In addition to full access to the "Market" section, the client gets access to the "Server" section. This is a fully ready-to-use product that requires nothing else. A robot fully automates market analysis, executes trades, and maintains the balance of currencies across all connected accounts. All trade parameters can be customized in the "Server" -> "Settings" section. This way, each user can make the service more convenient according to their individual needs.

«Choose plan» section

To get started, click on the “Change Plan” section.



After registration, the free plan is selected by default. With the free plan, only the “Markets” section is available. The “Pro” plan provide access to the “Markets” and “Server” sections.

Free

EUR **0.00*** / mo

~~10-EUR/mo~~

- ✓ Access to market, analytics, orderbooks sections
- ✓ Access to statistics
- ✓ Reporting
- ✓ Incoming chains list

Subscribe

Pro

-91%

EUR **15.00*** / mo







~~165-EUR/mo~~

- ✓ Access to market, analytics, orderbooks sections
- ✓ Access to statistics
- ✓ Reporting
- ✓ Incoming chains list
- ✓ API and Secret key are issued
- ✓ Processing of the received chains with external programs
- ✓ Server is provided
- ✓ Script for trading operations is provided
- ✓ Ability to work with internal API
- ✓ Support
- ✓ Faster chain generation

✓ **Activated**

15. PERSONAL ACCOUNT. «MARKETS» SECTION

This page lists exchanges where you can make transactions using our program. The “Average” and “Last” ping values show the average and most recent connection speed to the exchange (in seconds). These boxes also include information on exchange commissions, base currencies, and uptime.

ACCEPTABLE MARKETS					
 BTS Bitstamp	 BNC Binance	 BFX Bitfinex	 BTX Bittrex	 CEX CEX.IO	 KCN Kucoin
Average ping: 0.296 Last ping: 0.27 Uptime: 99.94% Commission: 0.3% ⓘ Base: btc, usd ⓘ	Average ping: 0.797 Last ping: 0.912 Uptime: 99.99% Commission: 0.1% ⓘ Base: btc, usdt ⓘ	Average ping: 0.359 Last ping: 0.326 Uptime: 99.69% Commission: 0.2% ⓘ Base: btc, usd ⓘ	Average ping: 0.125 Last ping: 0.134 Uptime: 99.98% Commission: 0.75% ⓘ Base: btc, usd ⓘ	Average ping: 0.533 Last ping: 0.388 Uptime: 98.55% Commission: 0.25% ⓘ Base: btc, usd ⓘ	Average ping: 0.644 Last ping: 0.646 Uptime: 99.96% Commission: 0.1% ⓘ Base: btc, usdt ⓘ

The number of bars shaded green, or orange indicates the current connection level to the exchange.

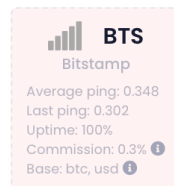
Excellent compound:



Straight-chain compound:

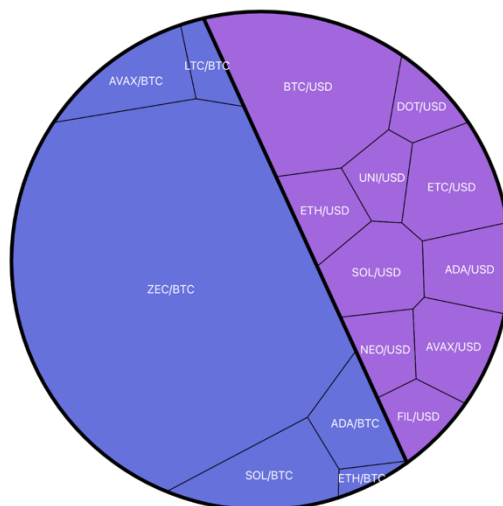


The exchange is disabled:



Pairs

This section begins with a cloud that shows the relative weight of pairs. You can choose the period you would like to display: day, week, or month. This graph clearly shows which pairs are the most used and profitable.



Below, you can also see an analysis of the pairs used on each exchange. This section displays the currently available pairs, highlighting the most used and least used pairs.

Available pairs

LTC/BTC ETH/BTC ETC/BTC ZEC/BTC XMR/BTC XRP/BTC EOS/BTC OMG/BTC TRX/BTC XLM/BTC VET/BTC XTZ/BTC BTC/USD ETH/USD LTC/USD
EOS/USD UOS/BTC CHZ/USD ADA/BTC ADA/USD FET/USD DOT/USD LINK/USD COMP/USD EGLD/USD UNI/USD AVAX/USD SNX/USD YFI/USD FIL/USD

[More](#) ▾

Most useful pairs

JST/USD BTC/USD ATH/USD

Not useful pairs

MANA/USD MANA/BTC UOS/BTC LINK/USD COMP/USD EGLD/USD AVAX/USD SUSHI/USD AAVE/USD SUSHI/USD NEAR/USD BOSON/USD DOGE/USD 1INCH/USD
FORTH/USD NEXO/BTC NEXO/USD WAVES/USD BCH/USD DOGE/BTC BCH/BTC MATIC/BTC MATIC/USD AVAX/BTC JASMY/USD SHIB/USD SPELL/USD THETA/USD
TRADE/USD CCD/USD GBP/USD GALA/USD BAT/BTC SAND/USD SWEAT/USD ETHW/USD BLUR/USD FLOKI/USD KARATE/USD PEPE/USD KAVA/USD TOMI/USD BONK/USD
MEME/USD APP/USD ZETA/USD CELO/USD STRK/USD AZERO/USD MPC/USD LYM/USD TOKEN/USD SPEC/USD PORTAL/USD TAIKO/USD BLAST/USD UXLINK/USD ZEC/USD
DASH/BTC DASH/USD IOT/USD IOT/BTC BAT/USD OMG/USD OMG/ETH

This table displays the available pairs involved in the chains. It is designed to facilitate the analysis of chains and pairs. The filter allows you to select any pairs you are interested in.

Most useful chains

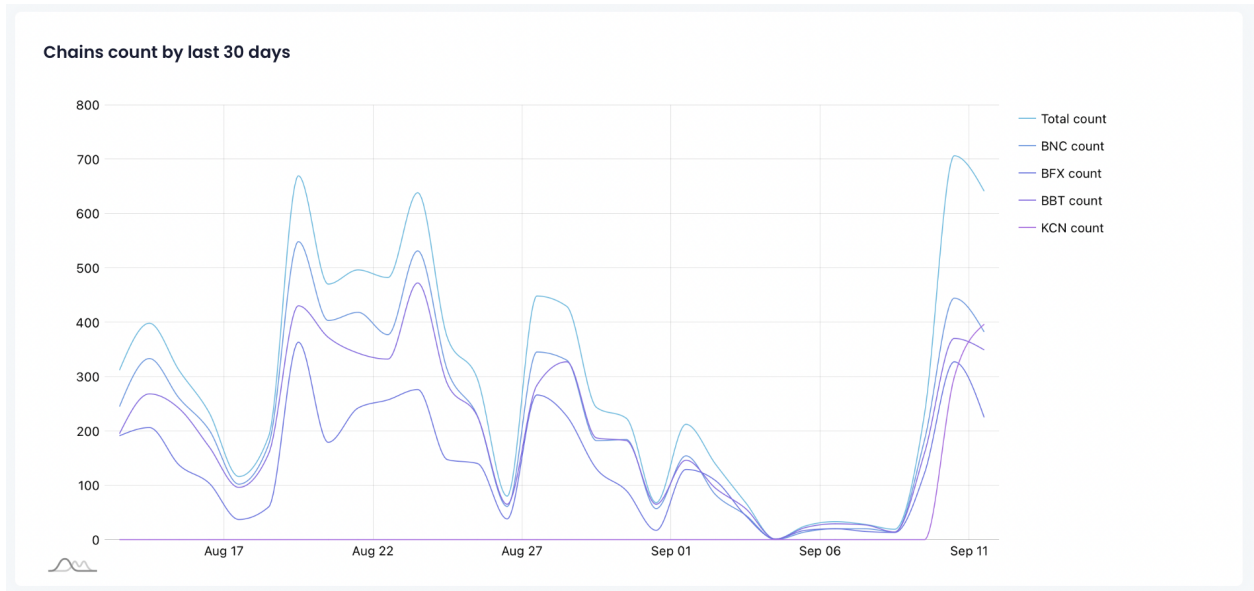
BTC/USD ▾

CHAIN	DEEP	APPEARED	PROFIT
BFX:ADA/USD - BTX:ADA/BTC - BTS:BTC/USD	3	15	58.18%
BTS:BTC/USD - BFX:ADA/BTC - BTX:ADA/USD	3	11	41.72%
BTS:AVAX/USD - BTX:AVAX/BTC - BTS:BTC/USD	3	4	34.80%
BTS:BTC/USD - BNC:AVAX/BTC - BTX:AVAX/USD	3	5	29.08%
BTS:ADA/USD - BTX:ADA/BTC - BTS:BTC/USD	3	7	22.16%

1. Analyzed chain;
2. Chain depth;
3. Number of times this chain appeared on the exchange;
4. Profit gained using this chain.

Analyzer

In the Analyzer section, you can view and sort the incoming chains according to your preferences. This section also features a chart that shows the number of chains received from each exchange for each day over the past month.



At the bottom of this page is “Get analyzer data” button.

Get analyzer data

If you click on the button, an additional pop-up window will appear. Here, you can select the exchanges and pairs you want to view. Then, choose the period for which you want to see the chains (no more than 7 days). Select the type of chains you are interested in and specify the minimum profit for the chains. Finally, click the “Submit” button and wait a moment. The process may take a couple of minutes.

Get analyzer data

Market

Bitfinex

Pair *

JST/USD

Period *

Pick dates

Reverse chain

☐ Show reverse chains

Full chain

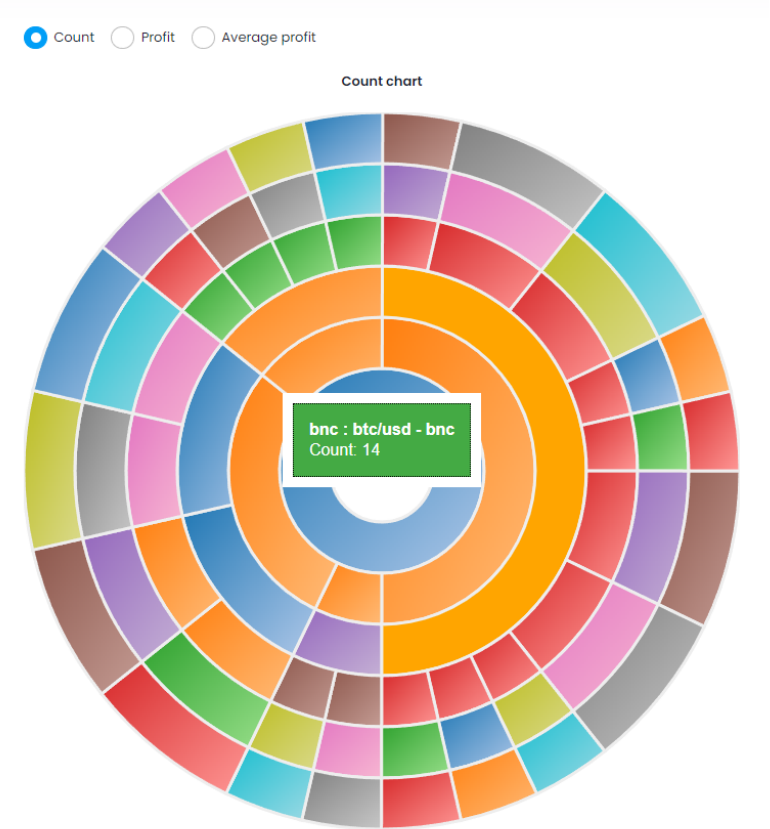
☐ Show full only

Minimal percent

0,2

Submit

Below is presented a circular diagram of currency pairs.



Here, you can choose the type of pair analysis. The larger the cell size of a pair, the more frequently it is used or the more profitable it is.

Count Profit Average profit

At the bottom of the page, you can see a table that displays data for both direct and reverse chains.

Analyzer List

Chain	Direct count	Direct days	Direct history	Direct profit	Reverse count	Reverse days	Reverse history	Reverse profit
bts:xrp/usd-cex:xrp/usd	5503	13	2022-03-31:6 2022-04-11:31 2022-04-21:1 2022-04-25:8 2022-04-30:19 2022-05-01:2 2022-05-09:5 2022-05-10:2 2022-05-11:3608 2022-05-12:1816 2022-05-13:1 2022-05-14:2 2022-05-16:2	2126.15	11	3	2022-05-09:5 2022-05-11:4 2022-05-13:2	14.74
bts:btc/usd-cex:btc/usd	1335	6	2022-04-18:7 2022-05-07:3 2022-05-10:1 2022-05-11:41 2022-05-12:1271 2022-05-13:12	576.28	911	6	2022-04-18:5 2022-05-07:3 2022-05-10:1 2022-05-11:41 2022-05-12:849 2022-05-13:12	402.27

1. Analyzed chains
2. Number of direct chains
3. Number of days when the number of chains specified in column 2 occurred

4. Number of chains per day
5. Profit gained from this type of chain
6. Number of reverse chains
7. Number of days when the number of chains specified in column 6 occurred
8. Number of chains per day
9. Profit gained from this type of chain.

Stream Full/Stream Filtered

This section displays statistics on various aspects of the program. The project offers two options for obtaining chains: Stream Full, and Stream Filtered.

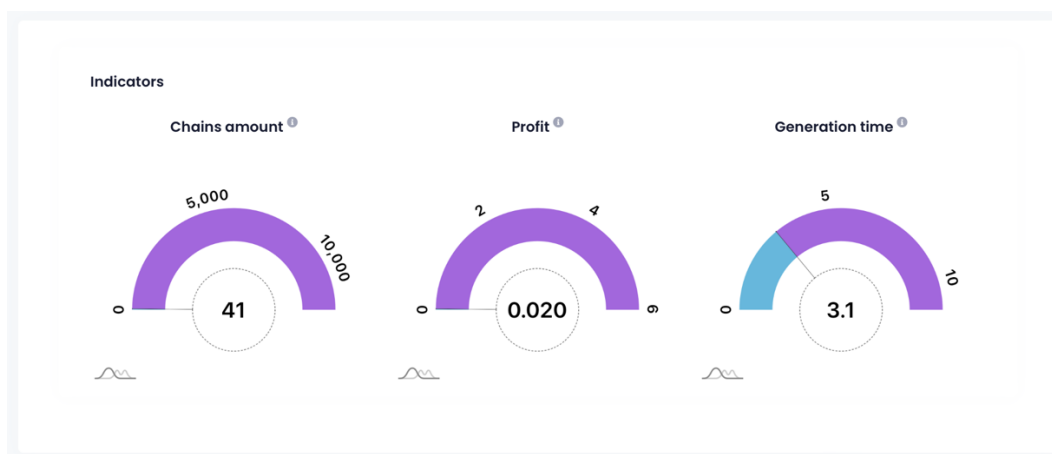
Stream Full: Data is received without filtering and comes as a continuous stream.

Stream Filtered: Data is filtered based on suitability for further use according to a set of criteria. Indicators display the current values of the last hour in relation to the maximum values:

Chains Amount: Number of chains received over the last hour compared to the maximum number.

Profit: Average profit over the last hour compared to the maximum average profit.

Generation Time: Average chain generation time over the last hour compared to the maximum average time.



This distribution shows the weight of each cryptocurrency pair with the corresponding exchange. The 'x' values represent the cryptocurrency pairs on specific exchanges (with exchanges abbreviated). The 'y' values reflect the frequency of the cryptocurrency pair appearing in the dataset over the past hour.

A bar chart displaying the distribution of normalized values for various pairs. The y-axis represents the frequency or count, ranging from 0 to 0.5. The x-axis lists numerous pairs, including kcmj/btc, bnc:btc/usd, bfx:xrp/usd, bfx:btc/usd, bnc:xmjbtc, bnc:xmjlusd, bbt:cnr/usd, bnc:cvc/usd, knn:btc/usd, bnc:kavajbtc, knn:fet/btc, knn:nij/btc, bnc:flux/usd, bfx:neo/usd, bfx:etc/usd, bbt:ftm/usd, knn:kava/usd, knn:dob/usd, knn:flux/usd, bbt:enai/usd, knn:chz/btc, bnc:aave/usd, bbt:akrol/usd, bnc:eri/usd, bfx:eur/usd, knn:twt/usd, knn:lray/btc, knn:near/usd, knn:chz/usd, bnc:totajbtc, bnc:zrx/btc, bfx:itc/btc, knn:aave/btc, and bnc:ada/btc. A vertical dashed line at approximately 0.01857 highlights the pair bfx:ada/btc, which is also indicated by a purple callout box.

Last hour market distribution chart

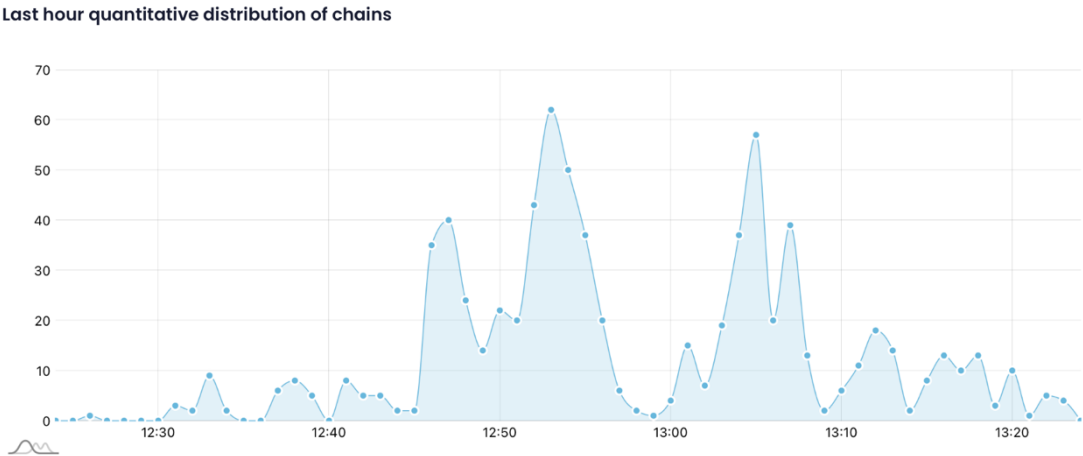
Segment Color	Value	Percentage
Dark Blue	0.34341	17.49%
Light Blue	0.34341	17.49%
Medium Blue	0.34341	17.49%
Dark Blue	0.34341	17.49%

Profit distribution diagram

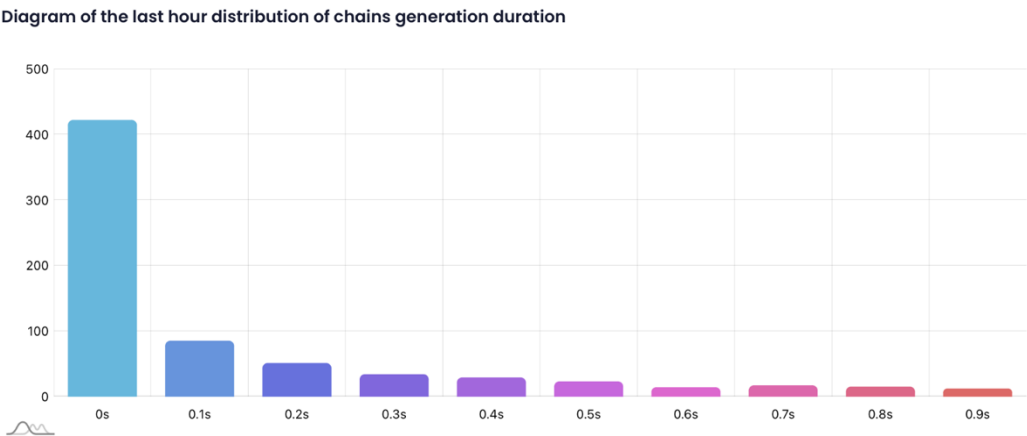
A bar chart titled "Profit distribution diagram" showing the distribution of profits across three categories. The y-axis represents frequency, ranging from 0 to 40 in increments of 5. The x-axis shows three categories: "<0.1", "<0.2", and "<0.4". The first bar, for the "<0.1" category, is blue and has a height of 35. A label above this bar indicates "<0.1: Profit: 35". The second bar, for the "<0.2" category, is a lighter blue and has a height of 5. The third bar, for the "<0.4" category, is a very light blue and has a height of 1. A small icon of a bell curve is located at the bottom left of the chart area.

Category	Frequency
<0.1	35
<0.2	5
<0.4	1

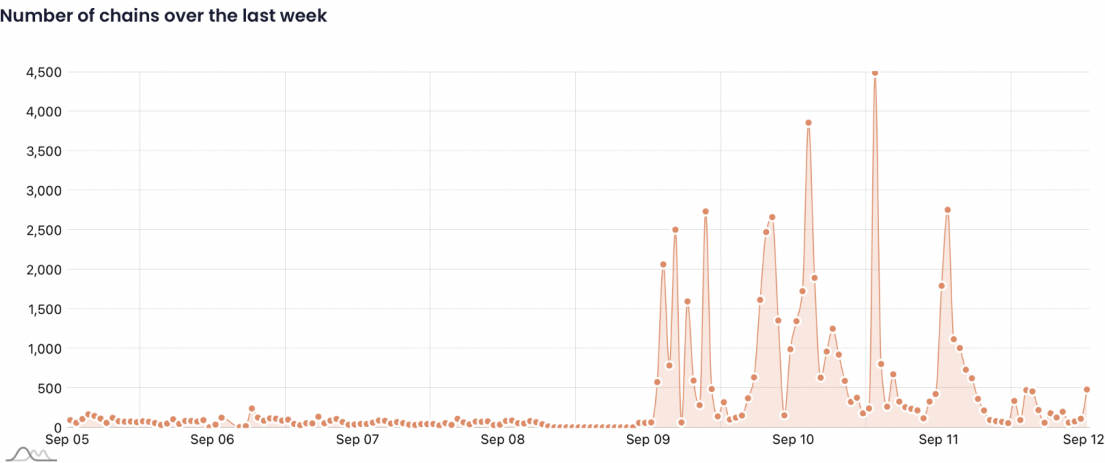
This distribution shows the number of chains received during a specific minute within the given hour. The 'x' values represent the time of chain receipt. The 'y' values represent the number of chains.



This bar chart shows the number of chains corresponding to a specific duration of generation. The 'x' values represent the duration of chain generation in seconds. The 'y' values represent the number of chains.

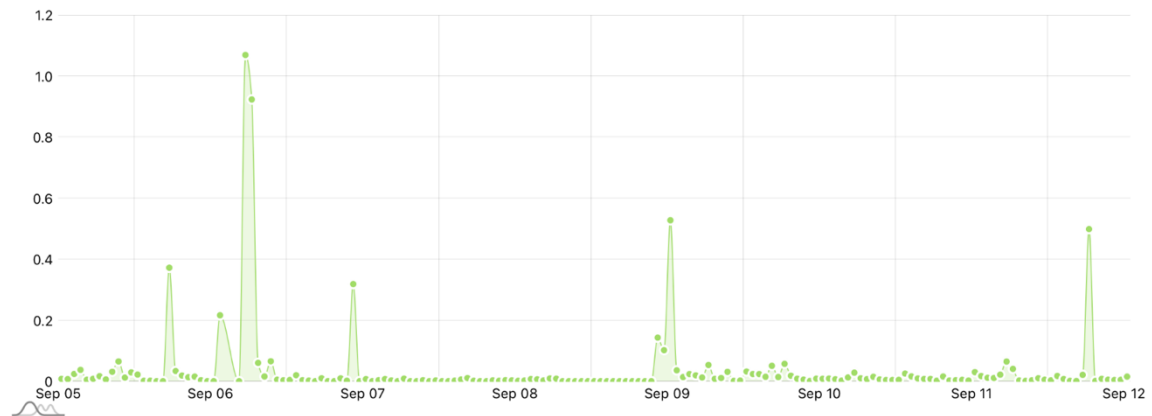


This graph shows the number of chains received during a specific hour over the past week. The 'x' values represent the time of chain receipt. The 'y' values represent the number of chains.



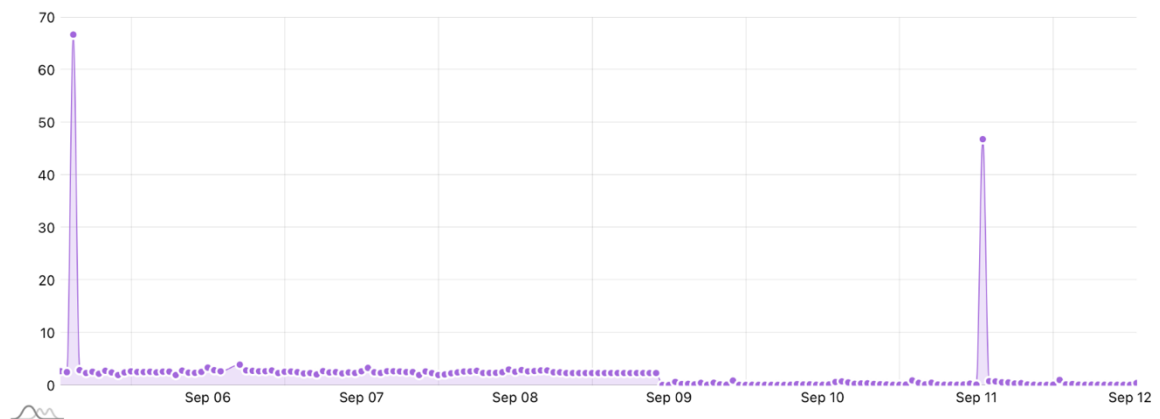
This distribution shows the amount of profit received during a specific hour over the past week. The 'x' values represent the profit amount for that specific hour. The 'y' values represent the amount of profit.

Profit distribution diagram over the last week



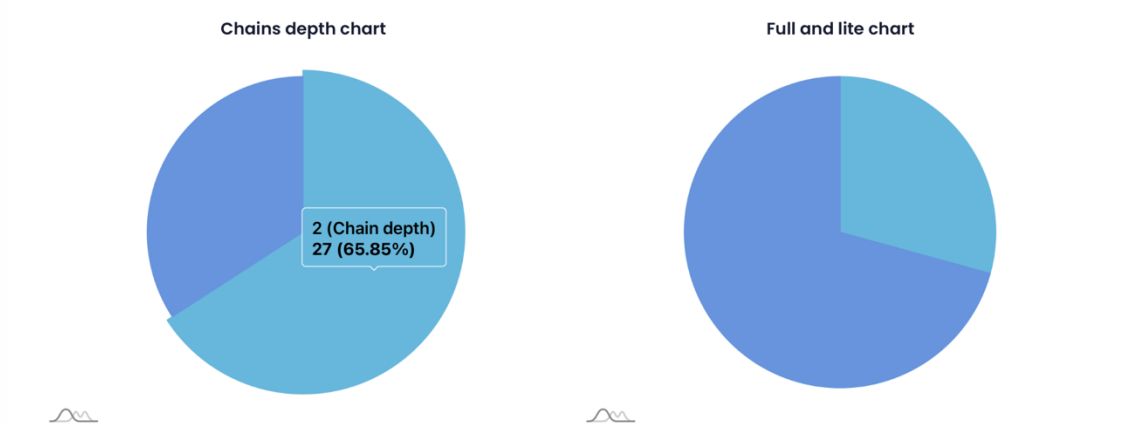
The graph shows the average time required to generate chains during each hour over the past week. The 'x' values represent specific hours for which the average generation speed is indicated. The 'y' values represent the generation speed in seconds.

Chains distribution timeline over the last week



The left pie chart shows the distribution of the number of chains consisting of two and three pairs. An example of a depth-2 chain (involving 2 pairs) is: bfx:trx/btc-bnc:trx/btc. Examples of a depth-3 chain (involving 3 pairs) are: bfx:eth/btc-bnc:sol/eth-bnc:sol/btc.

The right one shows the distribution of the number of full and lite chains. Lite - the transaction occurs at the best price, while the volume does not matter. Full - the deal passes at a weighted average price.



Below you can see the latest chains that are suitable for further use. These chains serve as the material for the program's ongoing work and are displayed in real time.

Last chains

Date:2022-01-01 01:58:15 | Chain: cex:btc/usd-bfx:dash/usd-btx:dash/btc | Lite: true | Profit: 100.00935737824419 | Spent time: 1.632
Date:2022-01-01 01:57:47 | Chain: bfx:dash/usd-btx:dash/btc-cex:btc/usd | Lite: true | Profit: 100.01676273606563 | Spent time: 9.187
Date:2022-01-01 01:58:08 | Chain: cex:btc/usd-bfx:dash/usd-btx:dash/btc | Lite: true | Profit: 100.01676273606563 | Spent time: 9.187
Date:2022-01-01 01:57:47 | Chain: bfx:dash/usd-btx:dash/btc-cex:btc/usd | Lite: true | Profit: 100.00195311694429 | Spent time: 7.893
Date:2022-01-01 01:58:01 | Chain: cex:btc/usd-bfx:dash/usd-btx:dash/btc | Lite: true | Profit: 100.00195311694429 | Spent time: 7.893
Date:2022-01-01 01:57:47 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.03841494929218 | Spent time: 1.441
Date:2022-01-01 01:57:47 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.0453406590744 | Spent time: 1.481
Date:2022-01-01 01:56:41 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.01764357199711 | Spent time: 1.745
Date:2022-01-01 01:56:41 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.02456640595258 | Spent time: 1.903
Date:2022-01-01 01:54:47 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.01659971644938 | Spent time: 1.396
Date:2022-01-01 01:54:47 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.02697631760967 | Spent time: 2.102
Date:2022-01-01 01:54:47 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.02351721132898 | Spent time: 1.499
Date:2022-01-01 01:54:47 | Chain: bfx:dash/btc-btx:dash/btc | Lite: true | Profit: 100.02005834428385 | Spent time: 1.237
Date:2022-01-01 01:38:32 | Chain: bts:xlm/usd-bfx:xlm/usd | Lite: true | Profit: 100.04675734814813 | Spent time: 1004.411
Date:2022-01-01 01:38:32 | Chain: bts:xlm/usd-bfx:xlm/usd | Lite: false | Profit: 100.00066818527404 | Spent time: 2.052

Orderbooks

You can select the exchange and pairs by clicking on them at the top of the page. These graphs display the order books (cups). The Ask is shown in red, and the Bid is shown in green. They are displayed in real time.



16. PERSONAL ACCOUNT. «STATISTICS» SECTION

This section displays the service's performance statistics. It contains four parts: trading statistics, workers status, quantum stacks, and a list of blocked pairs.

The "Statistics" block displays information about the service's current operational capabilities. "Markets" shows the number of exchanges connected to the service, while "Pairs" indicates the total number of pairs received from these exchanges. "Chains" represents the number of chains received. "Incoming flow" reflects the number of ticks per second coming from exchanges for processing. "Flow to Quantum Stacks" refers to the number of order books per second sent to GPU servers for calculations, whereas "Flow to Hetha" indicates the number of order books sent to the Hetha.io website. The next parameter represents the number of chains calculated on the GPU in the last second. The last parameter, generation time, denotes the average time required for generating chains.

Statistics

Markets: **5**

Pairs: **1176**

Chains: **14827**

Incoming flow (ticks/s.): **13477**

Flow to Quantum Stacks (ticks/s.): **2136**

Flow to Hetha (ticks/s.): **1908**

Chains calculated per second: **24934**

Avg gen. time (s.) **2.25**

This table presents data on Workers servers. Workers are servers that receive ticks, generate order books, and send them to GPU servers for chain calculations and to Hetha for further transmission. The Worker column lists the server names. The Active column indicates whether a server is on or off. Incoming flow shows the number of ticks per second coming from exchanges for processing. Flow to Bergelmir/Thrym/Kari 1 min displays the number of order books per minute sent to the GPU server for calculations. Flow to Hetha is the number of order books sent to the Hetha.io website.

Workers

		Flow to					
Worker	Active	Incoming flow 1 min.	Hetha 1 min.	Bergelmir 1 min.	Thrym 1 min.	Kari 1 min.	
Reginn	●	17670	55	902	130	371	More
Bomburr	●	39561	66	1744	78	2430	More
Brokk	●	30893	98	1210	251	938	More
Durinn	●	28008	124	3741	169	401	More
Gandalf	●	14755	54	1668	92	542	More
Alfr	●	34142	54	1270	131	1322	More
Total		418372	1192	23213	1913	15064	

[Show More](#)

If you click on the “More” button, you can see the number of ticks in and ticks out for the chosen server related to a specific market.

Reginn

		Flow to				
Market	Incoming flow	Hetha 1 min.	Bergelmir 1 min.	Thrym 1 min.	Kari 1 min.	
okx	7134	1589	0	0	0	More
bbt	7920	397	0	0	0	More
kcn	114388	9778	0	0	0	More
bnc	1581	427	0	0	0	More
bfk	3752	328	0	0	0	More

[Close](#)

Here, you can also see the “More” button. If you click on it, you will access the most detailed table, which displays ticks in and ticks out for a chosen market and a specific currency pair. Reason for Tick Rejection lists the reasons why ticks were discarded and not included in order book processing:

- R1 – The incoming tick is identical to the previous one.
- R2 – The difference between ask and bid is too small.
- R3 – Order book integrity is violated.

R4 – The price or volume of ask or bid is undefined or invalid.

R5 – The previous and current order books are identical.

Reginn okx										
Pair	Incoming flow	Flow to				Reason for Tick Rejection				
		Hetha 1 min.	Bergelmir 1 min.	Thrym 1 min.	Kari 1 min.	R1	R2	R3	R4	R5
velodrome/usd	350	49	0	0	0	13	0	287	0	0
parti/usd	577	60	0	0	0	24	0	486	0	0
core/usd	557	95	0	0	0	45	0	412	0	0
axs/usd	366	79	0	0	0	20	0	266	0	0
cro/usd	455	101	0	0	0	7	0	340	0	0
comp/usd	434	59	0	0	0	8	0	357	0	0
hbar/usd	557	245	0	0	0	20	0	281	0	0
stx/usd	477	147	0	0	0	34	0	291	0	0
zro/usd	514	107	0	0	0	53	0	338	0	1
foxy/usd	248	50	0	0	0	8	0	190	0	0

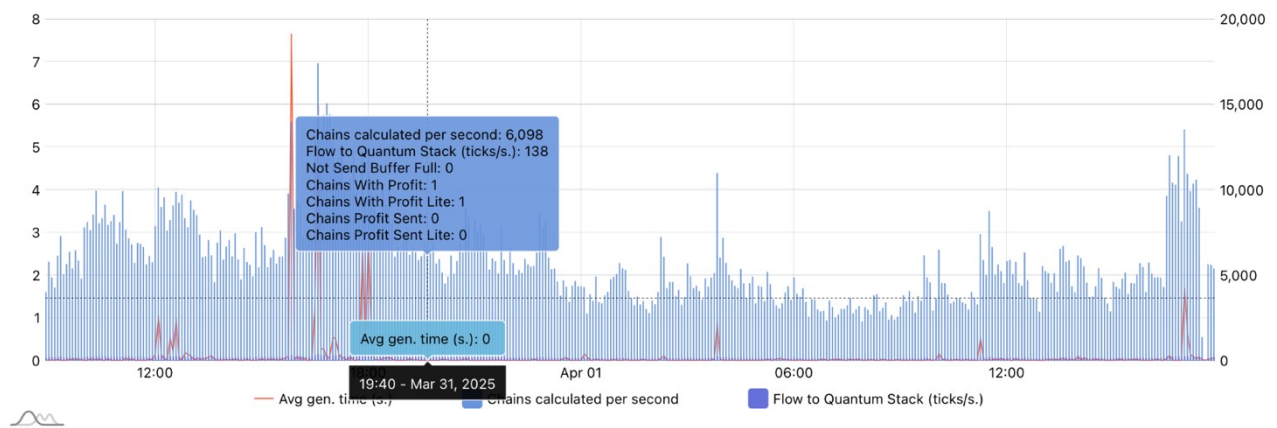
This table presents data on GPU servers. Quantum Stacks are GPU servers that calculate chains. The table provides data for different time periods: 1 minute, 10 minutes, and 60 minutes. Average generation time is the average time in seconds required to generate chains over a specific period. Chains amount shows the number of profitable chains generated and sent. Incoming flow displays the number of incoming ticks to the GPU server.

Quantum Stacks							
	1 min		10 min		60 min		1 min
	Avg gen. time (s.)	Chains amount	Avg gen. time (s.)	Chains amount	Avg gen. time (s.)	Chains amount	Incoming flow
Bergelmir	0.00	0	0.16	106	0.72	207	45834
Thrym	0.00	0	0.10	13	0.40	394	24228
Kari	0.00	0	0.06	4	0.23	58	26760
Total	0.00	0	0.11	123	0.45	659	96822

This graph presents data on GPU servers. Average generation time is the average time in seconds required to generate chains. Chains calculated per second shows the number of chains processed by the GPU. Flow to Quantum Stacks displays the number of order books per second sent to the GPU server for calculations. The graph has a 5-minute step; however, the values are divided accordingly to represent data per second.

Quantum Stacks Graphs

Bergelmir



Additional Information:

- Chains calculated per second – The number of chains processed by the GPU.
- Flow to Quantum Stack (ticks/s.) – The number of order books per second sent to GPU servers for calculations.
- Not Send Buffer Full – Chains were not sent because the buffer was full.
- Chains With Profit – Full profitable chains.
- Chains With Profit Lite – Lite profitable chains.
- Chains Profit Sent – Full profitable chains sent to Hetha.io.
- Chains Profit Sent Lite – Lite profitable chains sent to Hetha.io.

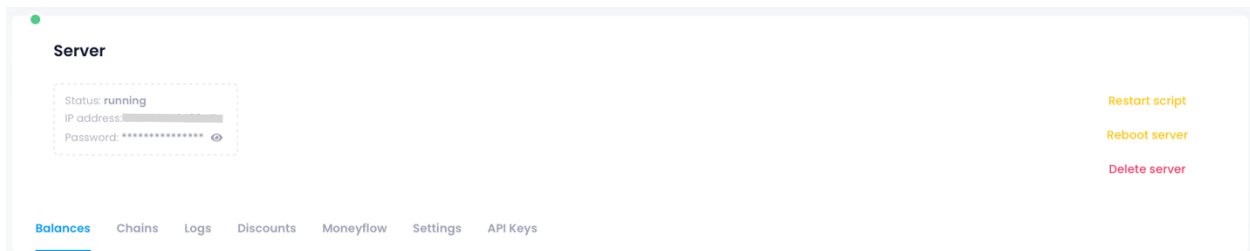
The last block, Blacklist currencies, displays the currencies that have been blacklisted by the service for various reasons. Chains involving these currencies are blocked.

Black list currencies

BTC3L AMP ACE NEIRO BIFI ALT HTX AI ZKX ZRX FET ENJ NMR MTL XNO CELR ZETA TST LAYER

17. PERSONAL ACCOUNT. «SERVER» SECTION

After starting the server, you need to wait a few minutes as the process may take some time. Once the server is ready, the indicator will turn green.



In the Server tab, you can manage your server, view logs, set discounts, and adjust the default settings.

To restart the script, click:

Restart script

To restart the server, click:

Reboot server

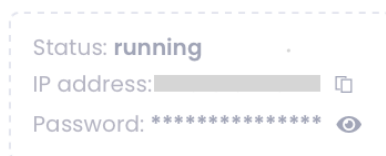
To delete it, click:

Delete server

If a new version is available, the "update" button will appear. To update the version, click:

Update server

Here you can see the status, IP address of the server and server password. Username: root.



Balances

This section displays the daily balances of various currencies. The main table shows the total currency balance across different exchanges. For detailed information on balance sheets from specific exchanges, click on "+".

By pressing "+", you can view the displayed currency balances on each exchange. The total balance is shown below. You can also see the exchange rate of each currency against the dollar. The balance sheet displays the USD equivalent of each currency. The last line shows the balance percentage ratio for each currency.

Chains

This table provides an analysis of the chains. It displays information on chains, profit, depth, forced status, multiplier, date, and execution status. To view a chain's history, click on the chain in the “Short” column. You can also select which profits and chain statuses to display here.

Show only open

Status

DONE

Comment

ALL

Delete chains

Type	Deep	Profit	Short	Status	Comment	Forced	Multiplier	Date
FULL	2	0.504	kcn:eth/btc-bnc:eth/btc	DONE close - bnc:ethbtc sell close - kcn:ethbtc buy		TRUE	8.00	2024-09-09 16:13:07
FULL	2	0.121	bnc:ada/btc-bfx:ada/btc	DONE close - bfx:adabtc sell open - bnc:adabtc buy		false	2.38	2024-09-05 10:33:10
FULL	2	1.352	bnc:eth/btc-kcn:eth/btc	DONE close - bnc:ethbtc buy open - kcn:ethbtc sell		false	23.71	2024-09-03 01:06:04
FULL	2	1.104	bfx:ada/btc-kcn:ada/btc	DONE close - bfx:adabtc buy close - kcn:adabtc sell		false	21.24	2024-09-02 07:47:36
LITE	2	0.268	bnc:ada/btc-kcn:ada/btc	DONE close - bnc:adabtc buy close - kcn:adabtc sell		false	2.00	2024-08-30 11:13:38
FULL	2	0.332	bnc:eth/btc-bfx:eth/btc	DONE close - bfx:ethbtc sell close - bnc:ethbtc buy		false	4.81	2024-08-27 14:10:46

1. Type of transaction

2. Chain depth

3. Gained profit (You can filter the profit you want to display using the “Comment” filter)

4. Chain composition

5. Chain status (filter available)

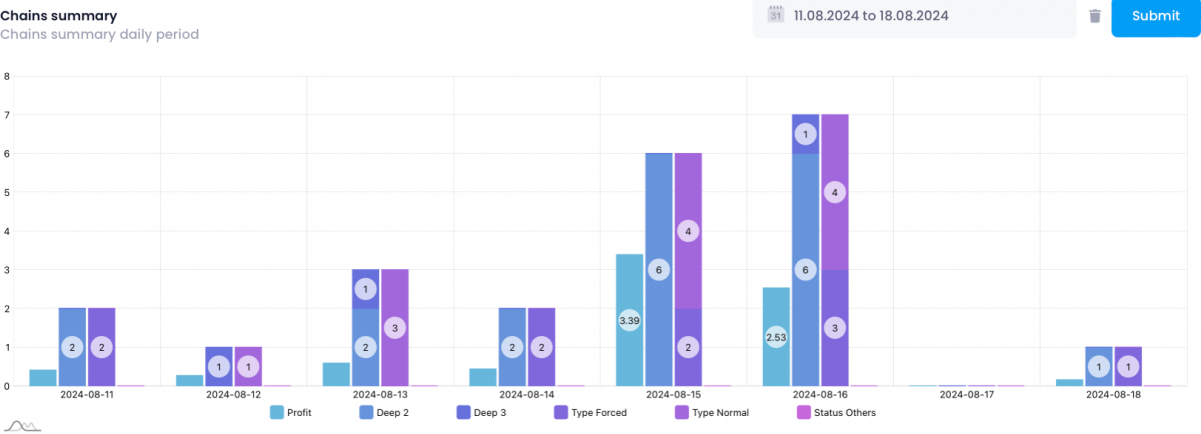
6. Comment (filter available)

7. Forced type (You can view the “Force log” by clicking the “True” button)

8. Increase the chain sum by “Multiplier” times relative to the base amount (You can edit this in the settings)

9. Date when this chain was profitable

At the bottom of this page is a bar chart. This chart displays a daily summary of chains, including profit, depth, forced and normal types, and status.



Logs

Here you can view the logs. To do so, select the date and time you are interested in. You can also find errors here; to view them, click on err.log. Additionally, there is chains.log, which contains information about the decision made regarding each chain. The “Logs” section displays information for the last 4 days only.

Logs

2023-03-26

2023-03-27

2023-03-28

2023-03-29

00.log

01.log

02.log

03.log

04.log

05.log

06.log

07.log

08.log

09.log

10.log

11.log

err.log

An example of logs is represented below:

Logs

2024-09-09

2024-09-10

2024-09-11

00.log

01.log

02.log

03.log

04.log

05.log

06.log

07.log

08.log

09.log

10.log

11.log

12.log

13.log

14.log

15.log

...

2024-09-11/ 02.log

Download full log

Search Logs

Search

2024-09-11 02:00:22.7 - 441343/127098880 : stats.js : =====

2024-09-11 02:00:22.7 - 441343/127098880 : stats.js : =====Previous hour summary=====

2024-09-11 02:00:22.7 - 441343/127098880 : stats.js : =====

2024-09-11 02:00:22.7 - 441343/127098880 : stats.js : hour:1

2024-09-11 02:00:22.7 - 441343/127098880 : stats.js : longPing: { "bbs":9,"bbs":1,"kcn":4,"bnc":4 }

2024-09-11 02:00:22.7 - 441343/127365120 : stats.js : Incoming chain, profit:

{ "0.02":188,"0.03":100,"0.06":14,"0.04":24,"0.07":14,"0.08":20,"0.05":17,"0.09":9,"0.10":2,"0.22":10,"0.15":7,"0.11":11,"0.13":12,"0.17":4,"0.14":8,"0.16":6,"0.30":5,"0.32":2,"0.33":1,"0.34":1,"0.28":7,"0.31":2,"0.23":4,"0.27":9,"0.29":9,"0.20":5,"0.21":4,"0.26":4,"0.19":2,"0.25":2,"0.18":5,"0.24":2,"0.12":8 }

2024-09-11 02:00:22.7 - 441343/127365120 : stats.js : Incoming chain, chain: { "kcn:avax/usc-bnc:avax/usc":3,"bbs:eth/usc-kcn:eth/usc":2,"kcn:btc/usc-bbt:ada/usc-bnc:ada/btc":1,"bbs:btc/usc-bnc:eth/usc-bnc:eth/btc":2,"bbs:btc/usc-bbt:ada/usc-bfx:ada/btc":6,"bbs:btc/usc-kcn:avax/usc-kcn:avax/btc":5,"kcn:btc/usc-kcn:ada/usc-bnc:ada/btc":3,"bbs:btc/usc-kcn:avax/usc-bnc:avax/btc":1,"kcn:btc/usc-kcn:avax/usc-bnc:avax/btc":1,"kcn:btc/usc-bbt:ada/usc-bfx:ada/btc":1,"kcn:btc/usc-bbs:ol/usc-bfx:ol/btc":1,"bbs:btc/usc-bfx:ol/usc-bnc:ol/btc":3,"kcn:btc/usc-bfx:trx/usc-bnc:trx/btc":6,"kcn:btc/usc-bfx:trx/usc-bnc:trx/btc":2,"kcn:btc/usc-bnc:ada/usc-bfx:ada/btc":3,"bbs:btc/usc-kcn:ada/usc-bfx:ada/btc":1,"bbs:btc/usc-bfx:trx/usc-kcn:trx/btc":7,"kcn:btc/usc-kcn:ada/usc-bfx:ada/btc":4,"bbs:btc/usc-bnc:avax/usc-kcn:avax/btc":5,"kcn:btc/usc-bnc:trx/usc-bnc:trx/btc":5,"bnc:avax/usc-kcn:avax/usc":6,"kcn:btc/usc-bfx:trx/usc-kcn:trx/btc":3,"kcn:sui/usc-bnc:sui/usc":2,"bbs:ada/usc-kcn:ada/usc":4,"bbs:eth/btc-kcn:eth/usc-bnc:btc/usc":11,"bnc:ada/usc-kcn:ada/usc":2,"bbs:ol/btc-kcn:ol/usc-bnc:btc/usc":2,"bnc:avax/btc-kcn:avax/usc-bnc:btc/usc":2,"bnc:ada/btc-bfx:ada/usc-bnc:btc/usc":1,"bbs:sui/usc-bnc:sui/usc":5,"kcn:btc/usc-kcn:avax/usc-kcn:avax/btc":3,"bbs:trx/usc-bnc:trx/usc":10,"kcn:btc/usc-bnc:trx/usc-kcn:trx/btc":3,"kcn:sui/usc-bfx:sui/usc":4,"kcn:ada/btc-bfx:ada/usc-

To find specific information, you can use the filter located at the top of the logs. The data entered in the filter is saved when you change the hour you are viewing.


Search Logs

Search

Discounts

Exchanges offer various commission discounts based on the use of their currencies or a specific turnover of funds. Discounts are personalized and can be manually adjusted in the table. You can also modify the price/profit ratio yourself. The "More" button opens a detailed breakdown of the discount based on turnover and exchange.

The table below displays the discounts available when using native currencies on each exchange. In the table below, you can specify discounts for more advantageous trading.

Markets	Discounts	Balance	Currency
BFX	<div><div></div><div>0,03%</div><div>%</div></div>	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	

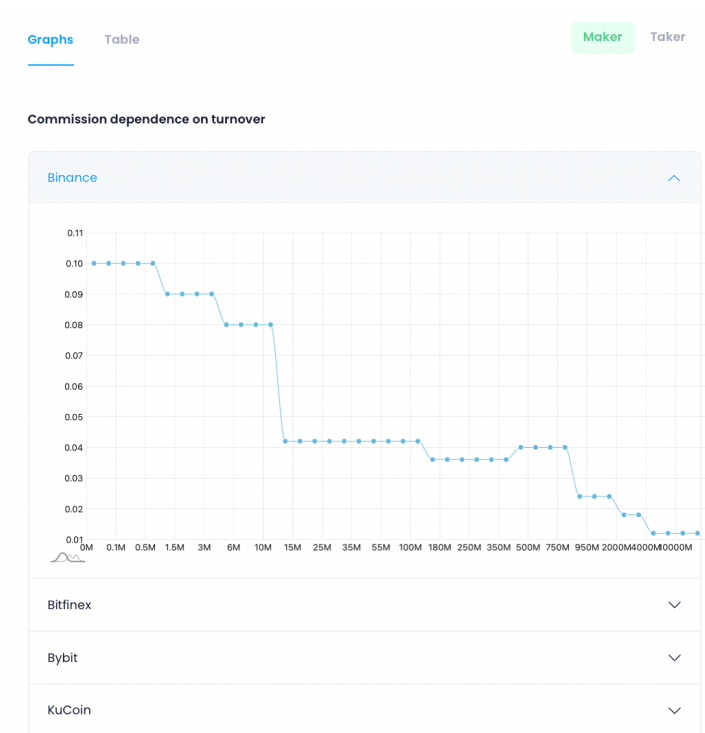
There are also two modes that you can enable or disable. By enabling the first mode, you can add a table with native currencies to your dashboard. The second mode allows you to activate or deactivate fee discounts.

Show native currencies ☒

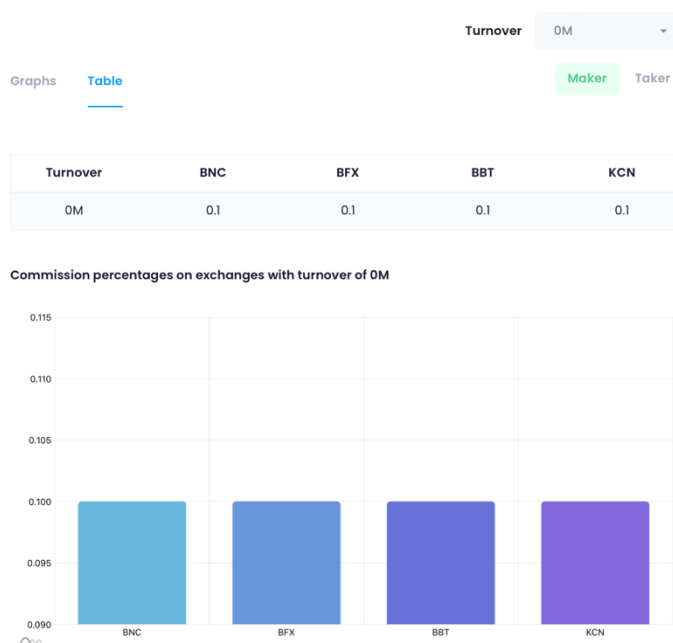
Activate fee discounts ☒

If you click on the “More” button, an additional pop-up window will appear. Maker - Commission percentage at execution of limited order. Taker - Commission percentage at execution of market order.

This “Graph” section shows commission dependence on turnover.



The “Table” section displays the specific commission rates for each exchange at various turnover levels. You can specify the turnover you are interested in at the top right of the window.



Moneyflow

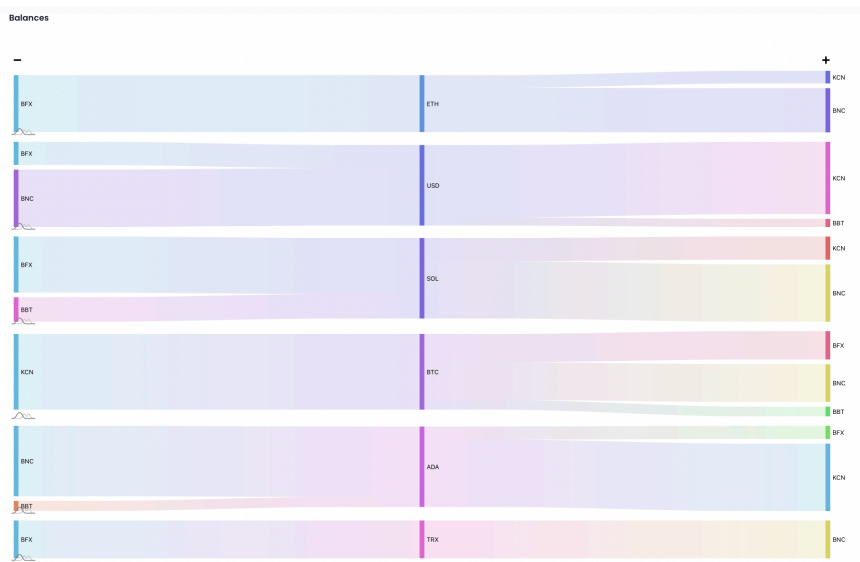
In this section, you can clearly view your moneyflow. This data covers a specific period of the month (e.g., September). You can choose the period you are interested in yourself.

At the top of the page is a table that displays the number of chains and trades. You can also see your turnover, the amount of your used assets, profit, return on capital and average profitability (as a percentage).

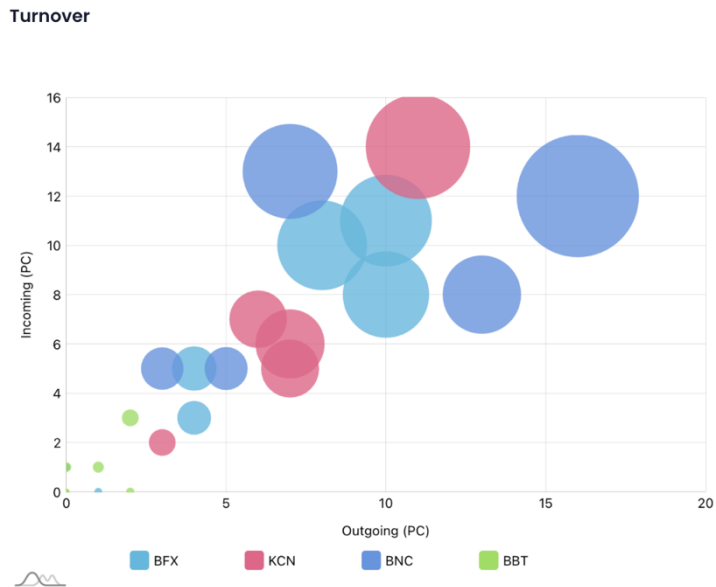
September 2025 Submit

Chains	Trades	Turnover (\$)	Used assets (\$)	Profit (\$)	Return on capital (%)	Avg. chain profitability (%)
4	8	800.00	400.00	2.08	0.52%	0.28%

Below the table, you can see a diagram that illustrates the money flow of each currency between exchanges. The thicker the line, the smaller the amount it represents.

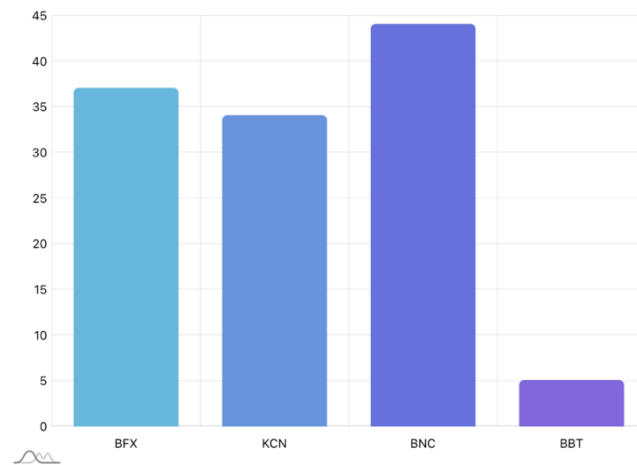


The next diagram displays the turnover of different currency pairs across various exchanges. The larger the circle, the higher the turnover.



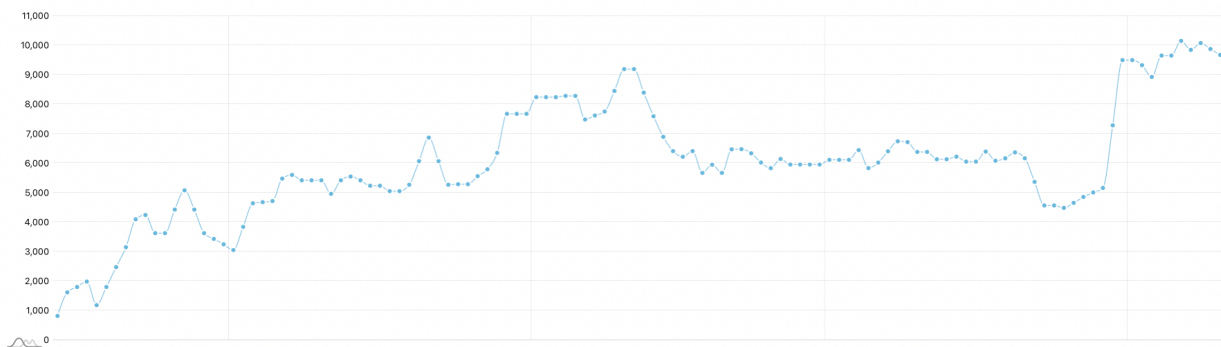
This bar chart displays the number of trades on each exchange.

Exchange trades distribution



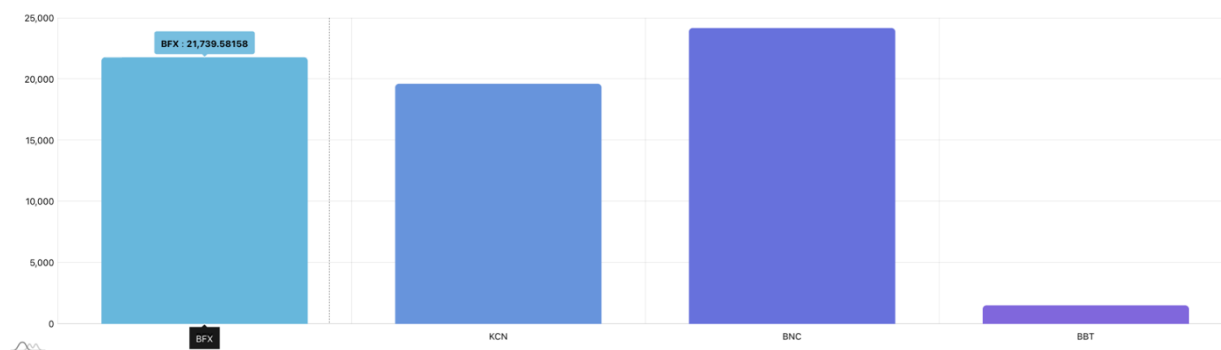
This graph illustrates the distribution of used assets for the specified month.

Last month used assets distribution timeline



This bar chart shows the turnover on each exchange.

Exchange turnover distribution



Orders

The period can be specified using the "Pick dates" filter and by clicking the "Submit" button. This filter specifies period for line line chart and tables.

Moreover, it is possible to enable/disable the “Show all opened” filter. If this filter is disabled, no data will be displayed in the “Opened” table.

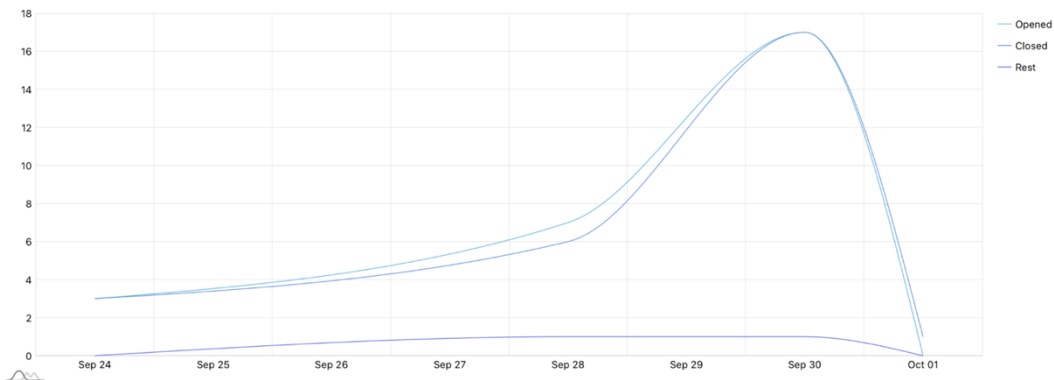
Show all opened

31

Pick dates

Submit

This line chart shows the number of opened, closed, and rest orders for each day during the selected period.



Below the chart, you can see tables with detailed data on open and closed orders. These tables include information such as generation time, chain and order IDs, exchange, pair, order status, and forecasted values.

Opened

Created	Chain id	Order id	Market	Pair	Side	Forecast
2024-09-03 01:06:04	1725325564648-719	66d660fcbc88be0007f33d0c	kcn	ethbtc	sell	-
2024-08-13 17:53:24	1723571604374-7010	656318130	bnc	trxbtc	buy	-

Closed

Created	Closed	Chain id	Order id	Market	Pair	Side	Forecast
2024-09-24 00:44:34	2024-09-24 00:44:34	1727138674162-8196	1015766328	bnc	adabtc	buy	base: 0.005925699602812529 pro: 1
2024-09-24 00:44:34	2024-09-24 00:59:54	1727138674162-8196	66f20b724c281b00073cbb05	kcn	btcbusd	buy	base: 0.028216905891895294 pro: 2661
2024-09-24 00:44:34	2024-09-24 00:44:34	1727138674162-8196	66f20b728f6d4b000746alc0	kcn	adausd	sell	base: 0.0005520605482161045 pro: 1
2024-09-23 14:39:33	2024-09-23 14:39:32	1727102372734-7700	30671910205	bnc	btcbusd	buy	base: 0.0010010291589424014 pro: 978
2024-09-23 14:39:33	2024-09-23 14:53:12	1727102372734-7700	66f17da5d083c4000768d89f	kcn	ethusd	sell	base: 0.00042858722736127675 pro: 1
2024-09-23 14:39:33	2024-09-23 14:39:33	1727102372734-7700	174270420654	bfx	ethbtc	buy	base: 0.02737412042915821 pro: 1
2024-09-23 13:48:28	2024-09-23 13:55:37	1727099305163-7328	4125116862	bnc	ethbtc	buy	base: 0.08632102608680725 pro: 1
2024-09-23 13:48:28	2024-09-23 13:48:28	1727099305163-7328	20275507374	bnc	ethusd	sell	base: 0.0014388785930350423 pro: 321
2024-09-23 13:48:28	2024-09-23 13:54:03	1727099305163-7328	66f17fac44c04400076b4be1	kcn	btcbusd	buy	base: 0.0005064045544713736 pro: 4
2024-09-23 08:20:27	2024-09-23 08:20:26	1727079626620-7139	174227719992	bfx	solusd	sell	base: 0.001210074673071504 pro: 43

10 Showing 1 to 10 of 18 records

1

2

API Keys

The "API Keys" section is where your keys are stored. Exchange and API keys are located here.

Api key

Key

Secret

upYYSQegTSxePAFOQxCYpoR91

.....

To start making transactions, you need to enter the keys for the exchanges where you will make transactions.

Bitfinex

Key *

Secret *

Discard

Save changes

If storing keys in your personal account is not preferable, you can enable "Local" mode, and the keys will be stored on the server.

Keys storage

Current keys storage: Base

Local key storage

Discard

Save changes

Settings

In this section, you can adjust parameters yourself. The table below shows the default settings, which you can modify by clicking on the respective fields.

MODULE	SETTINGS	VALUE	COMMENT
ROOT	lock_timeout	<input type="text" value="180000"/>	Unlock timeout
ROOT	percent_balance	<input type="text" value="0.9"/>	The amount of the deal cannot exceed the VALUE of the balance
LITE	accept	<input type="text" value="true"/>	Accept Lite deals
LITE	percent_accept	<input type="text" value="0.2"/>	Accept chain if profit is more than VALUE%
LITE	force_trading	<input type="text" value="3"/>	Use market orders if profit is more than VALUE%
LITE	rate_diff	<input type="text" value="0.05"/>	Accept chain if orderbooks difference is not more than VALUE%
LITE	multiplier	<input type="text" value="2"/>	Multiply chain amount in VALUE times
FULL	accept	<input type="text" value="true"/>	Accept Full deals
FULL	percent_accept	<input type="text" value="0.1"/>	Accept chain if profit is more than VALUE%
FULL	force_trading	<input type="text" value="1.5"/>	Use market orders if profit is more than VALUE%
FULL	rate_diff	<input type="text" value="0.1"/>	Accept chain if orderbooks difference is not more than VALUE%
FULL	multiplier	<input type="text" value="4"/>	Multiply chain amount in VALUE times

Do not forget to click the “Update” button under the table to save your changes.

Save changes

At the top right of this page, there are two buttons. By clicking on “JSON,” you can view these parameters in JSON format. You can also reset to default settings by clicking on “Default Settings”.

JSON

Default settings

18. PERSONAL ACCOUNT. OTHER SECTIONS

Downloads

The “Downloads” section includes a brief description of the program, as well as some commands presented as code snippets.

Get started

To execute the program, you must insert code:

```
const API = require('../libs/api.js');
api = new API( error );
```

Statistical data of the chain analyzer are available in a visual format in the user's personal account and can also be viewed by this request:

```
let command = 'report/chains';
let params = {"chain": "xrp", "dateFrom": "2020-08-15", "dateTo": "2020-09-24", "reverse": true, "profit": "100.2", "lite": "0"};
api.request( command, params, function ( response ) {
    console.log(response);
});
```

Using this command, you can save chains by yourself and display them in your personal account.

```
var params = [feed.lite, orderId, feed.history, feed.deep, feed.profit, feed.chain];
api.request('saving/chain', params, function ( response ) {});
```

The file [feed.example.js](#) contains an example of a request to the FEED server

To execute the example, you must set the following current parameters:

```
const apiKey = 'your-api-key';
const apiSecret = 'your-api-secret';
```


APPLICATION

Example of Developing a Trading Module

It is necessary to develop a module that will be fully compatible with the trader project. The project can be found at the following link: <https://github.com/Hetha.io/trader/tree/main>

If necessary, a live version of the project can be obtained for real-time use and testing.

Examples of already completed modules can be found at the following link: <https://github.com/Hetha.io/trader/tree/main/libs/modules>

In this folder, you can currently find modules for connecting to the corresponding exchanges: bbt.js – bybit, bfx.js – bitfinex, bnc.js – binance, bts.js – bitstamp, btx.js – bittrex, cex.js – cex.io, kcn.js – kucoin, okx.js – okex. Currently, there are already 8 modules written, but there is a need to expand this list.

Below is a list of tasks necessary for writing a module for the Kraken (Krk) exchange:

tc.js, trade console — helps with testing the blocks mentioned above. When writing a new block, it should also be connected to the console. To connect the module, you need to place the file (module) in the folder mentioned earlier. The parameters for the connection are specified in the `../libs/params.json` folder.

Running tc.js

node tc

Functions that should run in tc.js

- m.krk.getBalances(callback);
- m.krk.getOrder("adausd", "order_nr", callback);
- m.krk.limitOrder("btcusd", "buy", 10, 0.50000, callback);
- m.krk.limitOrder("btcusd", "sell", 10, 5.0000, callback);
- m.krk.marketOrder("adausd", "buy", 10, callback);

The data types passed in these functions must fully comply with those indicated above. API Key and Secret for Kraken access will be provided.

Brief Overview of Required Adaptations for Each Specific Function in the Standard Module,

Using the Example of the bnc.js File.

function bnc(params, debug, log)

The input parameters do not change. All object parameters should remain exactly as indicated in the original. This applies to every part going forward. If a specific functionality is not explicitly part of the connection, everything else should remain as is.

You only need to change the initialization of the Binance class to Kraken.

Below, this function loads trading pair data to properly handle rounding when placing orders. In the case of Binance, this is necessary, but in other cases, such as in bfx.js (Bitfinex), this functionality might not be required. It is important to verify whether this functionality is needed for Kraken or not.

bnc.prototype.setCurrentRate

Leave it unchanged.

bnc.prototype.getMarketName

Leave it unchanged.

bnc.prototype.getBalances

Change the function for obtaining balances from Kraken.

After obtaining the balances, the internal logic should be transferred to the new function. In particular, recalculating the spent_time and forming the output balance array. Since the implementation of this function differs greatly between exchanges, it makes sense to not limit the study to bnc.js alone. It is recommended to review modules for other exchanges as well.

For balances, we are only interested in AVAILABLE and the SPOT component. The output balance array should be passed in the response object. Accordingly, the structure should remain unchanged, as in the original modules, to ensure full compatibility.

In the Binance module, there is normalization of currency names. In each specific case, we need to check if normalization is necessary. The reason is that different exchanges may refer to the same currencies differently. For example, we work with USD in external systems, while exchanges usually work with USDT. Therefore, the balance on the exchange will be USDT, but it should be shown as USD in the balance output.

function pairRecover

This function adapts the trading pair from the external call to the corresponding name on the exchange.

For example, in the external system, pairs have the format "btcusd", where the pair is always written in lowercase and joined together. The second currency in the pair has a length of 3. However, pairs in different exchanges may appear differently: USDT-BTC, btc-usdt, etc.

It is necessary to determine how pairs appear on Kraken and, if needed, make the appropriate changes.

bnc.prototype.getOrder

The function must retrieve order data based on orderId. The retrieved data should be transformed into a response object, filling in the fields ID, PRICE (total order fill price), AMOUNT (order volume), and RESPONSE.

bnc.prototype.orderStatus

This function retrieves data about an order based on orderId. The retrieved data should be transformed into a response object, filling in the fields ID, STATUS (open/closed), UPDATE (last order update), and RESPONSE.

bnc.prototype.limitOrder

This function opens a limit order. You need to adapt the data for the corresponding exchange, such as PAIR, RATE, AMOUNT.

You should then call the limit order (buy or sell) using the exchange's interface. Afterward, pass the order data to the errorProcessing function.

bnc.prototype.marketOrder

This function opens a market order. You need to adapt the data for the corresponding exchange, such as PAIR, AMOUNT.

You should then call the market order (buy or sell) using the exchange's interface. Afterward, pass the order data to the errorProcessing function.

bnc.prototype.errorProcessing

Make adjustments to the filling of the corresponding fields. Each exchange has its own data format when returning data after placing an order. You need to extract the necessary data from the format and fill in the fields: ID, MARKET, PRICE, AMOUNT, RESPONSE.