TLDR

- Latency races will exist to some extent due to the nature of communication delay when participating in auctions and other transaction ordering mechanisms.
- Though sealed-bid block auctions have minimised the impact of these latency races, MEV teams will continue to optimise their latency and processing performance to find more arbitrage opportunities and produce more profitable bundles.
- Awareness through analysis, categorisation and research into these latency techniques within the on-chain trading lifecycle (trigger propagation latency, tick-to-trade, transaction propagation latency, supply chain latency) will help to democratise latency arbitrage opportunities as public infrastructure looks to adopt these optimisations.
- Some of these techniques (co-location, computer memory management, latency minimisation in a P2P network etc.) are ported from traditional literature (traditional finance, distributed systems networking etc.) whilst others are looked into on-the-go (relay protocol optimisation, block construction algorithms etc.) as more work is done to analyse the MEV supply chain.

Latency is a major driving factor still in the world of MEV (Maximal Extractable Value). As block times get shorter on newer chains, latency arbitrageurs and traders will continue to build low-latency trading systems that offer them more granular time precision in execution. Being able to pick on opportunities and execute them quicker in these contexts gives traders incentives to invest more (financially and effort-wise) into custom latency optimisations. On the research side, these optimisations become helpful in

pushing the field

forward such that other users on the blockchain can benefit from democratisation of these techniques. Building low-latency systems for on-chain trading and arbitrage is a multi-faceted problem. However, likening it to traditional trading systems helps to understand each part.

This article aims to analyse the operations involved trading on-chain and how it compares to the traditional HFT trading lifecycle; we also look to define a number of latency metrics that measure each part of this lifecycle, delving into where the segmentations lie and the key developments lowering these numbers further and further.

The Traditional Trading Lifecycle

Traditional

limit order book exchanges

host two-sided auctions that represent the market of some asset (with respect to the value of another asset). Sellers create asks for the price they want to sell at, while buyers create bids to buy at a certain price. When sellers and buyers cross, their orders are matched by the exchange's matching engine and a trade occurs.

Individuals or groups may participate within these auctions to speculate on the direction of the market and make profit. Strategies mechanically define how they may choose to capitalise on such opportunities, for which information is critical to reinforcing perceived trends or breakdowns to enter or exit a certain position.

Trading systems help to automate this process in code. They take in prices and other pieces of information to build an internal representation of the market (often an orderbook). From there, they will take note of key events (a new ask or bid listing, a new sale,

a market news headline

etc.) known as "triggers" that are used by a strategy (or strategies) to inform how the trading system should react. The system may generate some trade decision, be it an order placement or cancellation, that is then sent to the exchange venue to be executed.

To simplify, the steps involved can be bullet-pointed as follows:

- Exchange sends events to connected trading systems
- Trading system processes events to generate the set of actions it wants to perform on the exchange to maximise returns
- Actions are sent back to the exchange to be executed

The frequency of how often a trading system updates their positions characterise the style of a trader's strategies. High-frequency trading (HFT), as the name suggests, involves trading and competing on a high-frequency timescale, from milliseconds to microseconds -

the modal race to capture an arbitrage lasts 5-10 millionths of a second

. Market participants such as hedge funds may hold onto positions for months or even years as long-term investments, performing this trading lifecycle infrequently. Regardless of the length of the expected holding period, speedy execution using a trading system can minimise the risk of

price slippage

.

The Trade Settlement Process of Cryptocurrencies

Cryptocurrencies are becoming attractive assets to speculate on,

interesting

traditional finance firms to find ways to enter the market. Investors can have exposure to these digital assets through indirect investment products such as

funds

, however, one may choose to trade these assets directly for autonomy and control. Doing so subjects one to the settlement or consolidation process that comes with handling any type of asset. Traditional exchanges use regional clearing houses such as the NASDAQ or the New York Stock Exchange in the US to facilitate the finalisation of trades on stocks; the CME clearing house facilitates the finalisation of trades on their commodities futures and options contracts.

For cryptocurrencies that are being moved about on-chain, trades are subject to the digital settlement process of the blockchain, of which the blockchain acts as the intermediary of all trades. Transactions that involve some transfer or trade of assets are propagated across a peer-to-peer network of nodes known as the blockchain network. These nodes run some

standard software client

that facilitate networking and transaction processing. To establish a source of truth in the history of transactions, a consolidating node, service or entity is chosen to be the block proposer, either temporarily or indefinitely, to batch received pending transactions into a block and add it onto the blockchain.

A block proposer holds a similar responsibility to that of a traditional matching engine for an exchange, however, the proposer is given full control over which transactions are included within a block and how they are ordered. Usually, a matching engine will process transactions sequentially (first-come-first-served) as a level of fairness. Initiatives such as

proposer-builder-separation (PBS)

aim to offer opportunities that deter the block proposer from utilising its monopolising power in unfair ways that

discriminate for or against certain entities

. Transactions may often need to be routed through additional infrastructure first to enable these schemes.

The On-Chain Trading Lifecycle

Trading systems involving the trading of cryptocurrencies follow a similar flow to those that exist in traditional finance. They do so by using events that may occur on the blockchain network to understand the preparation, execution and settlement of transactions.

- Trigger propagation:

A node (or broadcaster) propagates either a pending transaction received from a user or a block of transactions proposed as the block proposer to connected peers and

systems

. This triggers the strategy to know when to run.

- Strategy execution:

Trading systems process these events to generate a set of operations/transactions they wish to perform on the blockchain to maximise (potential) returns.

- Transaction propagation:

Transactions are propagated to a nearest node within the blockchain network or to a transaction gateway service e.g. a

PBS-supported builder, an auction decider within OFAs etc. so that they can then be further propagated until it reaches the next block proposer.

- Supply chain propagation:

Once captured by a peer node or a transaction gateway service, the transaction is subject to the rest of the supply chain of operations (validations, processing, aggregations) before it is finally received in some form by the next block proposer.

The Importance of Minimising Latency as an On-Chain Trader

Latency can be defined as the time delay between the cause and the effect of some system: a trading system might not receive information in time to react optimally to a trigger event and prices can slip against a trader's favour if the execution of a trade is delayed. Other market participants that are able to react and settle their trades sooner can exploit arbitrage opportunities that would have otherwise gone to the trader. It is

approximated that traders in traditional finance markets lose 0.005%

from not being able to cancel their existing orders in time due to speedier arbitrageurs that snipe their stale quotes.

Latency is still a factor to consider when trading on-chain. Though the design space of

fair transaction ordering mechanisms

has been

explored

in detail to minimise the types of arbitrage opportunities that arise by the strategic ordering and transaction injection by a block proposer, there still exists some responsibility on a trader to make transactions as soon as possible to place, update or cancel existing orders promptly as a part of their trading strategy so that it is not picked off by others. Active participation is required within

priority gas auctions

to express preference but missing participation in these auctions due to latency can lead to worse price execution.

Defining Performance Metrics

To remain competitive, trading firms will monitor latency as one of their key performance indicators (KPIs). There are many sources of latency that exist in traditional trading lifecycles:

- Exchanges have to validate orders sent from traders. Is the user authenticated? Are negative prices allowed? Does the data need to be preprocessed for the matching engine?
- The method of connection (communication protocol) with the exchange can add some overhead if more messages than needed are having to be sent for the same data
- The types of operations on the hot code path can also be analysed to find ways to minimise the active computation that can only be done after a trigger event has been received

Similar sources of latency can be identified in blockchain-based systems as metrics to optimise. Below, I define four such metrics that exist within the on-chain trading lifecycle:

- Trigger propagation latency:

the time it takes for a trigger event to be sent from a node, block proposer, or off-chain service, and then received by the trading system.

- Tick-to-trade:

a metric borrowed from traditional finance. It measures the time it takes for a trading system to receive the trigger and then generate the trade decisions (transaction or transactions as a bundle) to execute.

- Transaction propagation latency:

the time it takes for a transaction or bundle to be sent from the trading system and received by the next block proposer (pre-PBS), the block builder (PBS), the rollup sequencer (layer 2) or auction decider (

OFAs

).

- Supply chain latency (PBS):

the time it takes for a transaction or bundle, upon receipt by the builder, to be validated, included in the built block, and then sent across a relayer to the block proposer. This metric is recorded from the time of receipt by the builder to when the proposer receives the block.

The following sections delve into each type, breaking down their definitions and what they entail, before highlighting some potential remedies that are generally used or are being researched into.

Trigger Propagation Latency

Trigger propagation latency is generally defined by the approach of propagation of the trigger event from the source system to one's trading system and how long this journey takes. A trading system will not be able to do any critical work until it has been alerted to the trigger event.

Events on the blockchain network, be it a new block or a new pending transaction, are propagated through a peer-to-peer gossip layer, which is how a node made aware of this new event. In a peer-to-peer network, nodes broadcast information through dispersing it to their direct peers in a undirected fashion, who then go on to broadcast themselves to their direct peers and so on, in accordance to the gossip protocol known to them. The global "internet" peer-to-peer network is very alike to the structure of a blockchain network.

For a message to be transmitted from one device to another, it must be sent through some physical communication channel where it must travel some physical distance. The most common approaches are through electromagnetic wave transmission (radiowaves, microwaves) or through cable wiring (fiberoptic or hollow-core fiber etc.) due to their speed and reliability.

It is clear to see that

being a direct peer to the original broadcaster of a trigger event is greatly beneficial

to minimise one's latency. Unless it is

otherwise as defined in the gossip protocol

, a node located within the same region as the original broadcaster is likely to be a direct peer, and thus, a crypto trading firm may choose to co-locate their nodes to static peers or place them in hotspot locations to hear first of the trigger event.

Several companies (such as

bloXroute

,

Blocknative

and

Chainbound

) host

dedicated block delivery networks

to systematically minimise the number of peers on average that have to receive the new block or transaction before it is streamed direct to one's trading system, sometimes

up to 2 seconds faster

. Supposing that a node only has the computational resources to be connected to 20 other nodes, it only takes 5 to be connected to 100 other nodes by ensuring that connections are unique. Additional techniques on the networking layer include

transaction caching

, the dropping of high latency connections and relying on other peers to route a transaction through, reducing

latency by 33% within simulations

.

Though it is possible to collude with the original broadcaster of a trigger event for first priority delivery rights, optimisations elsewhere are being looked and explored at the software layer of a node that consistently lower the time of delivery of the trigger event.

- Using other communication protocols than RPC to deliver received events can result in faster times. In traditional finance, this has lead to the development of trading-specific communication protocols such as

ITCH

and

FIX

.

- The filtering of trigger events from blocks and propagating only these events can save on transmission time.

Decker and Wattenhofer

found that the average propagation of a block on the Bitcoin network was about

2 seconds plus another 0.08 seconds per kilobyte

in the block. Propagating only important or compressed details can save on time.

- Software clients

may often have to verify incoming trigger events or messages containing trigger events before re-broadcasting or claiming receipt of this event. Verification generally happens as a part of the gossip protocol, however, it can be seen as redundant if all nodes in a trusted network have to repeat this process for whitelisted senders.

- Embedding a trading system into the client of a node eliminates an additional latency that would have otherwise been experienced to deliver the message of the trigger event to an external trading system. This is very alike to embedding trading systems onto network interface cards such that an extra network hop does not need to be made.

Firms are currently building their own custom blockchain clients behind closed doors due to the proprietary nature of the optimisations. However, we may see some attempts be built in the open, aiming to democratise the latency arms race. Large backing behind such initiatives could prove to be fruitful in making such speedy access to data accessible to all.

Tick-to-Trade

Depending on the complexity of a trading strategy, some processing will need to be done to either explore or solve within the space of transaction operations that can be chosen to provide some optimal execution.

Evaluating the latency involved in such methods and others for a trading strategy relies on analysis of the “hot path”. The hot path of a trading strategy is the code that is ran immediately upon receiving the trigger event “tick” to then generate the applicable set of actions or “trade” operations to perform. The definition of the tick-to-trade metric is borrowed from the traditional finance context, referring to how quickly one’s trading algorithm is able to react to a “tick” that comes in as a determining signal to then generate an applicable “trade” that we can send out to be executed.

As an example, sandwich attacks focus on observing pending user transactions within the mempool and utilising the gas-based bidding mechanism of the blockchain to place a transaction before and a transaction after the user’s to best arbitrage surrounding that event.

One potential strategy is as follows: if the user is deciding to buy token Y with token X on Uniswap V2, a transaction placed before to buy token Y with token X and then a transaction to sell token Y back into token X after can prove to be profitable. An optimisation problem lies here in finding the right amount to buy and then sell back. Implementations such as libevm’s

subway

and Supercycled’s

cake_sniper

use an iterative binary search method to find this optimal amount, but sometimes,

equations

may exist that can solve for the solution in one go to remove redundant computation.

Optimisations for the execution of one’s trading algorithm vary from programming language to programming language. Using performant languages as close to assembly such as C++ and Rust gives a programmer control over how memory is managed, allowing for more efficient execution. Each language has various resources available online for how a program can be made more performant, ranging from techniques such as

alternative implementations for standard data structures

to

optimistic caching

.

One major source of latency contributing to tick-to-trade latency is often not from the internal logic processing involved but rather calls to external components such as a web API or a simulation engine. A call to retrieve some data induces a round-time-trip (RTT) costing in latency that could be very time-expensive to stall and hear back from. If a trading system is trading bets on the weather in Dallas in the form of

Weather Futures contracts

, a strategy could involve predicting the real price using current global weather data and trading upon that. One way that this could be done is call an

external weather API

over the internet as every single tick of price data from the Weather Futures market come in (latency could range from 300ms - 2s, or more from geographically distanced locations like a call from New York to a data center in Singapore), however, this would not be appropriate for a high-frequency trading system looking to enter and exit trades every 100ms or sooner. This same example can be applied to blockchains with very short block times that set these constraints. Caching / pre-fetching the data from a web API to store in a local cache helps to move this processing off the hot path.

Some strategies may also opt to capture additional signals such as (

historical

) blockchain events to reconstruct the state of a protocol locally in wait. As the indexing space matures as a whole, it may be possible to see developments in data-as-a-service providers (

Alchemy

,

Transpose

,

The Graph

,

Blocknative

etc.) offering custom indexing capabilities that proactively transmit this data to subscribed clients, saving on half of the round trip time. Another possibility is the development of blockchain clients suited to custom indexing that can be ran locally.

A trading system may also need to work with a simulation engine, doing so on-the-fly; hence, calls to such a service will generally lie on the hot path and is thus performance-critical. Projects such as

Ganache

,

Hardhat Network

and

Anvil

provide the means of setting up such an environment for blockchain simulations, however, connection to these instances (even if local) induce a network cost. However, for others, solutions would likely converge to embedding simulation logic within the trading system (or vice versa) to save on the round time trip cost altogether. Such an example can be seen

here

using

revm

.

Transaction Propagation Latency

After a transaction, set of transactions, trades, or outputs etc. have been generated from a trading system, they are to be sent to a necessary recipient that can consolidate the trade and (hopefully) guarantee settlement. As with trigger propagation latency, transaction propagation latency is driven by the approach of delivery, however, routing will generally be directed towards a certain destination.

For centralised exchanges, this is the centralised matching engine server that drives all matching and settlement of crossed orders from an orderbook. It makes sense to co-locate trading system instances with the server to minimise the time it takes to propagate a trade intent. The same argument can be applied to

blockchains

or systems with a centralised sequencer or trade consolidator, where a direct connection often minimises the latency to deliver a transaction or trade.

Given the lack of disincentives in a P2P network, it is generally an

effective strategy

to broadcast a transaction to all possible peers as soon as possible to maximise the chances of using the shortest latency route. Though a shorter geographical distance generally correlates to a shorter latency, it is not always the case that the most direct route is the least latency-inducing and thus a

flooding strategy

helps to decrease the average transaction propagation latency even further. This is also an optimal strategy in systems where the trade consolidator is chosen in a non-deterministic fashion (take, for example, the Ethereum proof-of-work blockchain).

The destination of where a transaction should be routed to can change as infrastructure changes the flow of a trading lifecycle. Co-location with validators on the Ethereum network was originally an effective strategy for traders and arbitrageurs (which we call searchers) as they would receive transactions and build blocks themselves. With the large adoption of mev-boost (and indirectly, PBS) to shift the responsibility of block building from a validator to other entities, it then made sense to co-locate with the block builders instead.

As different services come to pioneer, change and revolutionise certain trading lifecycles on-chain (for example,

Skip Protocol

,

Jito Labs

,

DFlow

etc.), each will have their own gateways of receiving trades from a trader: a block builder (PBS), a sequencer (L2), an auction decider (OFAs) etc.

- Based on the architecture involved, it is possible for custom delivery networks to be implemented should transaction propagation latency be a concern e.g a custom delivery network for delivering to

Polygon Fastlane

, a delivery network that has direct co-located connections to the Arbitrum sequencer etc.

- In the existence of multiple instances of a gateway for a trading lifecycle (e.g

block builders

in the Ethereum PBS flow), a flooding strategy as per before can be applied to maximise the chance of execution. More advanced network designs such as

Perigee

and

Peri

may be explored in the future for smarter routing, finding techniques as similar in impact to pruning high-latency peers.

Greater assumptions (e.g trust, network architecture etc.) may be taken on to pick out rarer optimisations as the protocols behind certain blockchain networking layers are examined more deeply.

- As mentioned previously, the communication protocol that is used to deliver messages to the transaction gateway is another area of optimisation, where custom protocols other than RPC could be used. Data compression is another layer that can be done to reduce latency by bandwidth limitations.

Supply Chain Latency

After the submission of a trade or transaction to a gateway (block builder, sequencer etc.), the gateway will propagate it up a supply chain of validations, processes and aggregations before it is then settled on-chain by the block proposer.

The trader generally has no control over the time it takes for this trade or transaction to be propagated and settled on-chain. These components are generally considered out of their reach and thus they suffice to the mercy of the rest of the supply chain. They do have a choice over which gateway they use to have the trade settled on, especially if there are other competitive avenues. Though one could have the fastest and most direct connection to an exchange, the delay from the exchange receiving an order to the order going live (which is supply chain latency) adds an uncontrollable risk that prices can slip beneath one. FTX was noted as one of the

"slowest" centralised exchanges

to trade on, with it taking "150 milliseconds typically" to see a trade go live. In comparison, Binance would only take "5 to 10 milliseconds".

In the context of block auctions for on-chain trading, a trader must account for supply chain latency to ensure that they are aware of the latest time (on average) that the gateway can receive their order so that it can be received by the block proposer and included within the next block. In a

study done by Metrika

, 96% of the proposed Ethereum blocks analysed were not the ones with the highest rewards built by block builders; thus, knowing how components further up the supply chain operate allows one to minimise the chance of their transaction being missed within a block.

Supply chain latency for on-chain trading may be attributed to various factors, depending on the trading lifecycle involved. Under PBS, the journey from the builder receiving a transaction or bundle (multiple transactions) and constructing a block, to the validator receiving it in time to decide on the next block to propose would be measured as such. This will vary for different lifecycles (OFAs, L2s, traditional exchanges etc.) based on the processes involved.

Supply chain latency within proposer-builder-separation primarily comes from three sources:

- The delay involved in block construction before it is sent off
- The latency involved in the delivery of the block from a builder to the relay
- The latency involved in the delivery of the block from the relay to the proposer

A

block builder

has the responsibility of collecting transactions from users and constructing blocks that pay the proposer an optimal amount of gas fees. Transactions may come at various points of a block auction, but there exists a latency delay between sending a block to a block relay and the block proposer receiving it on the other end that a block builder must consider. Though block builders aren't

as driven to invest in latency

as more favourable dimensions of advantage exist, they may be driven by the priorities of the traders that send their exclusive orderflow to them as they become more receptive, including latency.

Co-location of the builder next to the relay (and relay next to the block proposer) is a given in offering latency benefits, but research has also delved into how the protocol for block submission could be adapted to reduce this further. One

proposal

from

Mike Neuder

,

Justin Drake

and

AlphaMonad

considers an optimistic relay that can submit blocks ahead of time to the proposer without awaiting for validation.

Though in its infancy, further developments to optimise latency in the PBS supply chain may soon be explored or implemented.

- Block builders will have different ways that they implement their block construction, some more scalable than others in dealing with larger throughputs. As blockspace demand grows,

stress testing

can help show the limits of how well a builder's algorithm is able to cope. Techniques can be used to optimise

public decentralised builders

.

- The simplification of messages and round-time-trips involved in the block submission protocol like that seen in the

optimistic relay proposal

can help with reducing physical latency. Getting rid of the relay is one approach but this loses benefits such as DDoS protection and dynamic routing to block proposers (that would then have to be done by block builders themselves).

Conclusion

Though the trading lifecycle on-chain is subject to the batching mechanism of the underlying blockchain that the assets are settled on, projects are building various pieces of infrastructure to impact the lifecycle in ways that democratise MEV opportunities. Latency is still a factor that is explored by both independent and institutional players to minimise price slippage. By finding nuances within blockchain systems that allow high-frequency traders to benefit from shorter latencies, traders do not have to express their full desire as a bribe within priority gas auctions, thus earning additional price improvements.

By analysing the lifecycle and defining a possible segmentation of the latencies involved, these latencies can be examined more closely. Techniques can be capitalised on by traders or democratised to give other market participants and users of the blockchain network fast execution. Time will tell how public infrastructure develops as the MEV space gets more competitive on all fronts.