

One Trading and AWS: Cloud-native colocation for crypto trading

by Atiek Arian, Boris Litvin, Hani Masri, Sercan Karaoglu, and Dr. Stefan Blackwood | on 15 DEC 2023 | in [Financial Services, Industries](#) | [Permalink](#) | [Comments](#) | [Share](#)

Latency is one of the sources of competitive edge and a continuous focus for exchanges and Market Makers in capital markets. Improving latency positively impacts the execution of trading strategies, enhancing liquidity and increasing profitability for market venues and participants. A significant number of crypto exchanges have been built on Amazon Web Services (AWS)—continuing to innovate to optimize network performance and offer cloud-native colocation.

In this blog we explored the topic of network latency in crypto trading and the advantages provided to exchanges and Market Makers by using shared [Amazon Elastic Compute Cloud \(Amazon EC2\) cluster placement groups](#) (CPGs). We will share representative testing performed in partnership with [One Trading](#), who have recently launched a new crypto exchange product that offers native colocation on the AWS Cloud.

AWS worked with One Trading, pre-launch, to quantify the expected Market Maker experience from the perspective of network latency across a range of potential exchange access topologies. We built a simplified high frequency trading (HFT) client that implemented a specific strategy and measured round-trip times for simulated orders, including matching engine latency.

About One Trading

One Trading's vision is to bring the highest standards of the traditional finance world to digital asset trading through technology and regulation. They have spent the last two years pursuing a regulatory strategy that will enable them to develop products in an increasingly competitive industry and build F.A.S.T., their next generation crypto digital asset trading venue for spot trading and, soon, regulated derivative products.

On working with AWS specialists during their pre-launch activities, One Trading said "We decided to work closely with AWS, looking to hit the types of latencies usually seen in the market leading traditional trading venues, but we wanted these to be available for all customer types whether they are retail or institutional. We embarked on an aggressive plan to bring our round-trip latency to under 200 microseconds.

"Our goal was to build a trading venue that could provide the fastest price discovery, fastest execution and be able to maintain this without performance degradation whatever the volumes thrown at the venue. Our ambition is to scale this product, using the AWS Cloud, beyond digital assets and under our new licensing structure we will be able to offer traditional securities for all customers."

Testing network architectures for exchange access

For testing purposes, a variety of network topologies were used to simulate real-world connectivity options that One Trading can provide, for exchange access, to their customers. Each topology provides a different latency profile and can be considered as different "connectivity tiers". A simulated Market Maker AWS Account was created within which we deployed the test HFT client. Primary emphasis was placed on optimizing network latency for these tests, hence only a single AWS Availability Zone (AZ) was in scope when measuring latency. The same AZ is used across both

Exchange and Market Maker accounts for all tests, to prevent additional network latency incurred from crossing AZ boundaries. Resilient architectures would involve the use of multiple AZs with leader-follower Amazon EC2 instances and synchronous or asynchronous replication mechanisms, usually implemented at the application layer. Our following diagrams depict a resilient configuration in two AZs.

Connectivity Tier-1: Amazon VPC peering with shared Amazon EC2 cluster placement groups

For this test configuration, an [Amazon Virtual Private Cloud \(Amazon VPC\) peering](#) connection was created between the One Trading VPC and the Market Maker VPC. An Amazon EC2 CPG was then created in the One Trading AWS Account that was shared with the Market Maker AWS Account. Trade engine, order gateway and matching engine Amazon EC2 instances were launched from both AWS Accounts into this CPG.

Amazon EC2 instances launched into a CPG benefit from greater locality on the underlying physical AWS network within the Availability Zone. Logical connectivity is achieved by using the peering connection and is private with traffic being carried over the AWS network in the AWS Region.

This topology provides for the lowest latency access to the exchange.

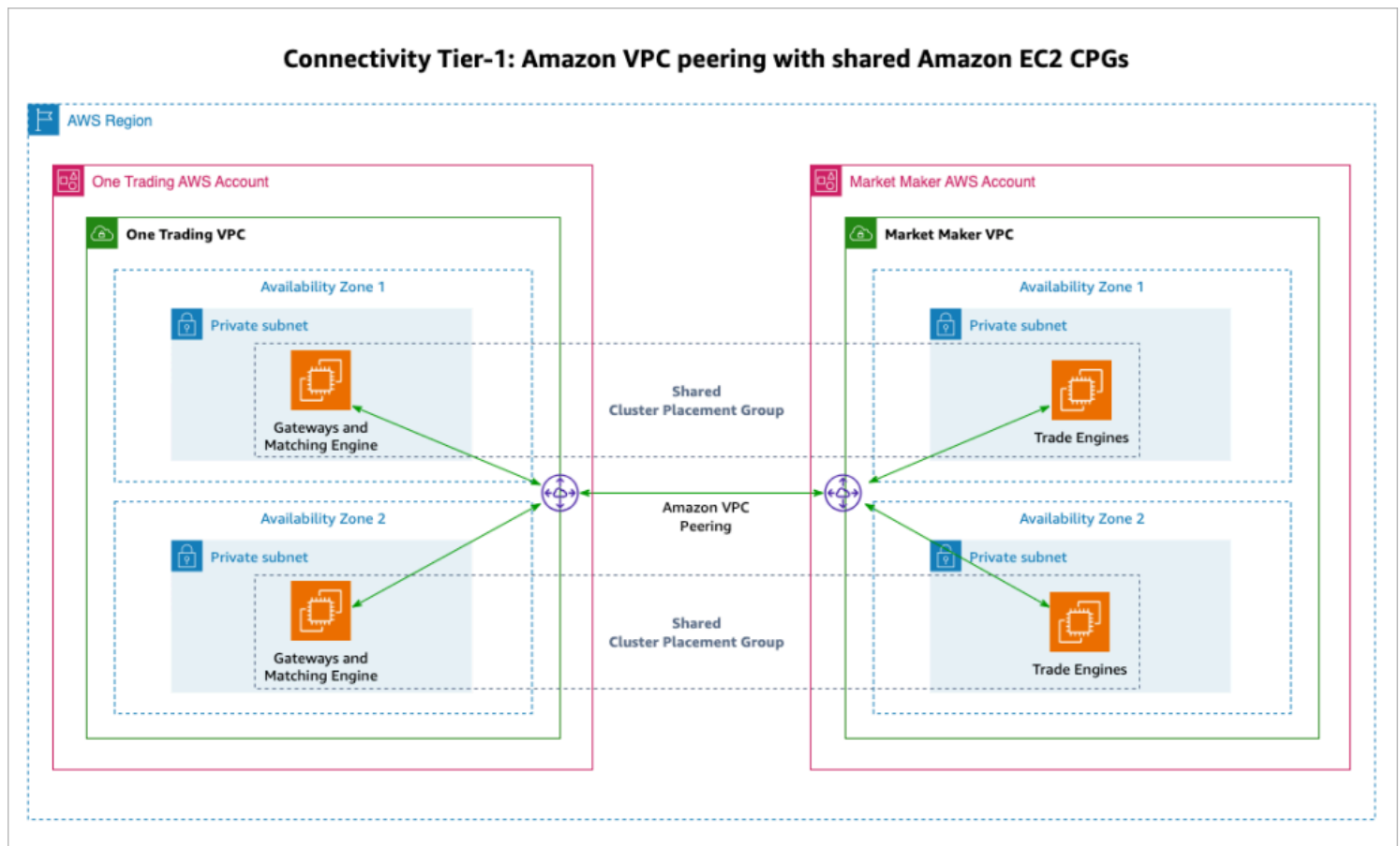


Figure 1. Connectivity Tier-1: Amazon VPC peering with shared Amazon EC2 CPGs

Connectivity Tier-2: Amazon VPC peering

This test configuration is similar to the previous, without the use of Amazon EC2 CPGs. It incorporates the default Amazon EC2 instance placement strategy where instances are randomly provisioned from the perspective of underlying physical capacity in the Availability Zone. Hence this does not include any special provision for network

locality between Amazon EC2 instances. Logical connectivity, as before, is private and achieved through Amazon VPC peering.

This topology provides for low, but not the lowest, latency access to the exchange.

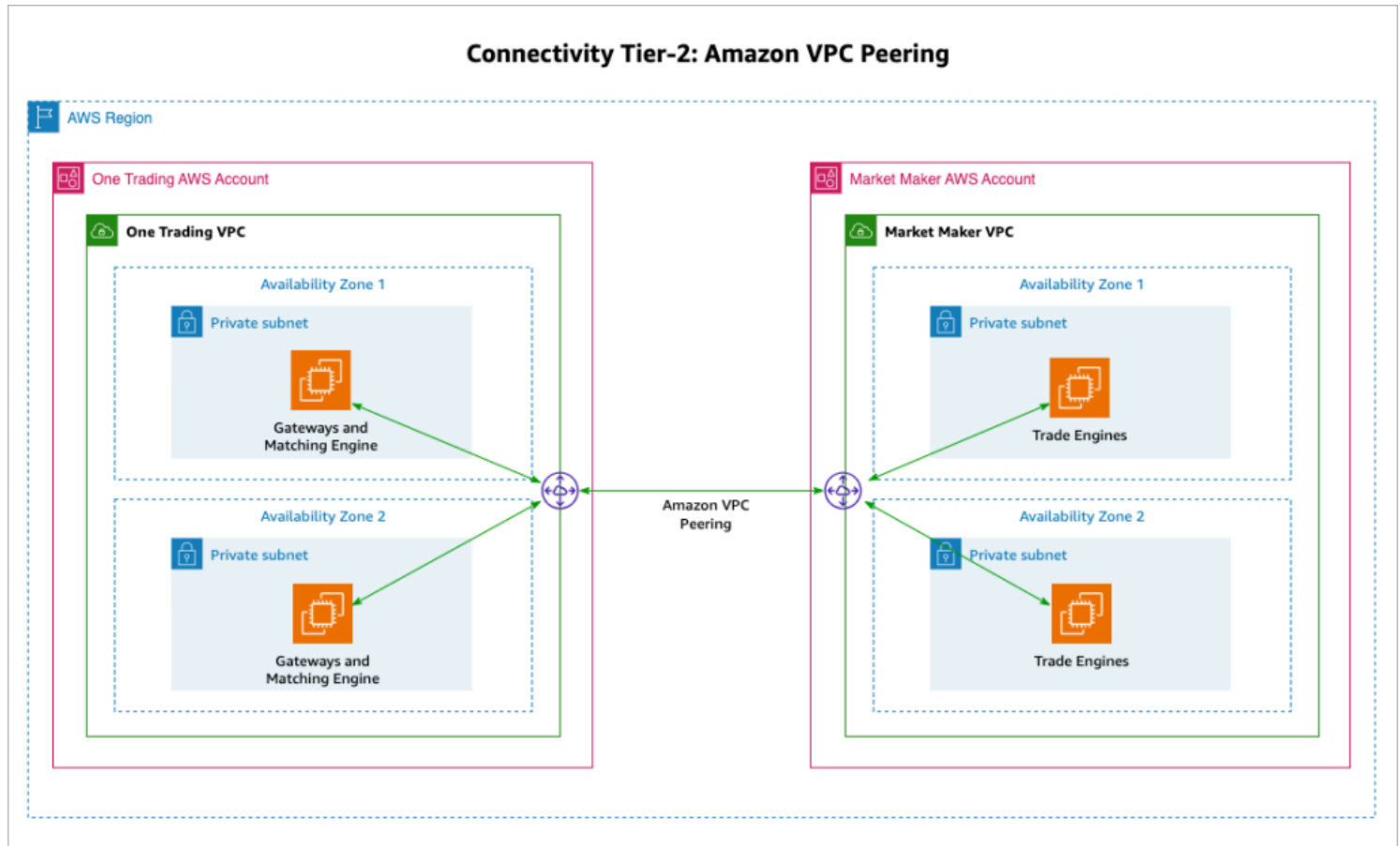


Figure 2. Connectivity Tier-2: Amazon VPC peering

Connectivity Tier-3: AWS PrivateLink

For this test configuration, logical connectivity between One Trading and Market Maker VPCs is provided by [AWS PrivateLink](#). PrivateLink is a fully managed service designed to provide connectivity between VPCs at extremely large scales (thousands of VPCs) and introduces additional hops in the network path (endpoints and load balancers).

This network topology continues to provide private connectivity over the AWS network within the AWS Region. PrivateLink requires the provision of an [Elastic Load Balancer](#)–Network Load Balancer inside the One Trading VPC through which the order gateways are exposed as a service. These gateways receive order flow traffic from PrivateLink endpoints created in the Market Maker VPC.

This topology provides for medium levels of latency access to the exchange.

Connectivity Tier-3: AWS PrivateLink

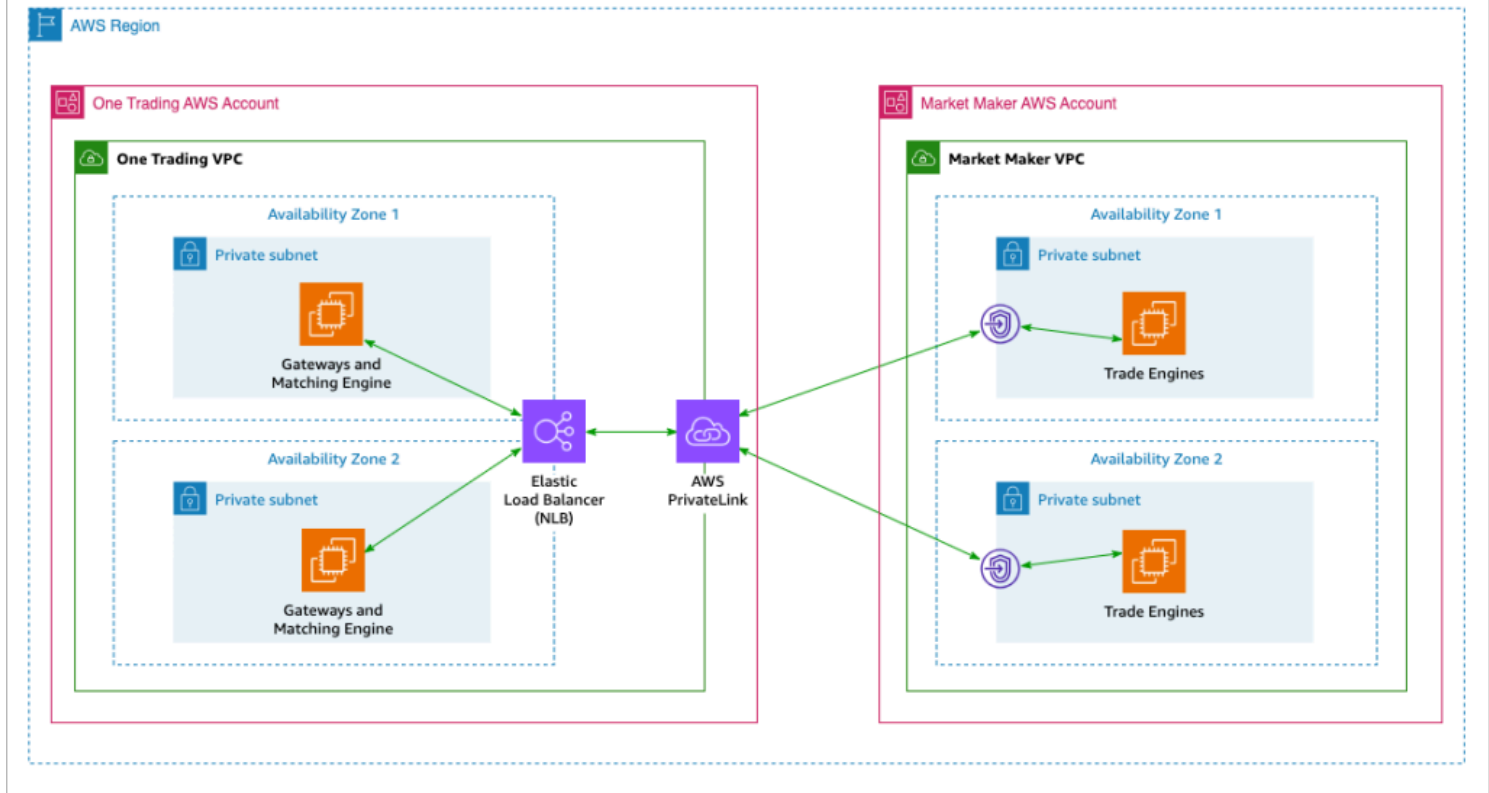


Figure 3. Connectivity Tier-3: AWS PrivateLink

Connectivity Tier-4: Internet through Amazon CloudFront

In this test configuration, connectivity is provided through an [Amazon CloudFront](#) content distribution network. The origin for the CloudFront distribution is a public facing Elastic Load Balancer which routes traffic to the order gateways in the One Trading VPC. This network topology involves public connectivity and allows for exchange access from both inside and outside the AWS Region.

Market Makers with footprints inside the AWS Region will achieve optimal connectivity since it is routed over the AWS Region and border network, and does not egress to the public Internet—despite traffic using a public IP space. Market Makers external to AWS can access the exchange by using their own Internet connectivity and will be routed to the nearest CloudFront point of presence (PoP).

If trade engines in the Market Maker VPC are required to be deployed into private subnets, this architecture can be supplemented by the addition of managed [NAT Gateways](#) in those VPCs. Do note that this would add additional latency.

Among all test configurations, this topology provides for the highest levels of latency access to the exchange.

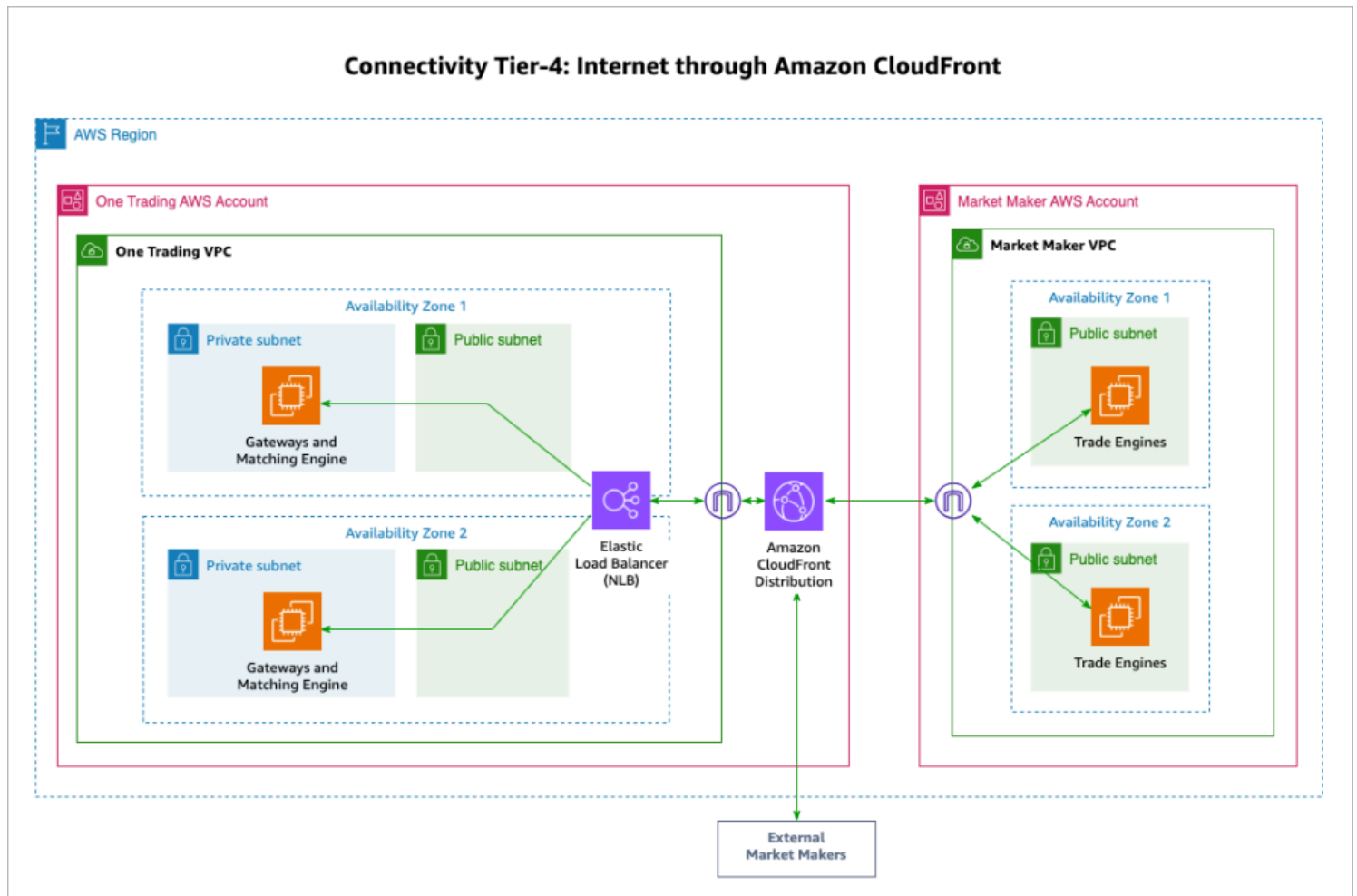


Figure 4. Internet through Amazon CloudFront

Testing methodology

We created a HFT client that was deployed on Amazon EC2 instances inside the Market Maker VPC. This client implemented a simplified low-latency trading algorithm that directed order flow, across the various network test topologies, to the One Trading exchange by executing the following steps:

1. The HFT client sends a limit order to the exchange. This order specifies a security and price, along with the direction (buy or sell).
2. The exchange generates an order acknowledgement (Ack) which confirms that the order has been received and is in the queue for execution.
3. The HFT client receives and processes the order acknowledgement. In response an order cancellation instruction is immediately sent back to the exchange.
4. The exchange generates an order cancellation acknowledgement.
5. The HFT client receives and processes the cancel order acknowledgement.

The following (Figure 5) is a simple sequence diagram illustrating the preceding message flow steps.

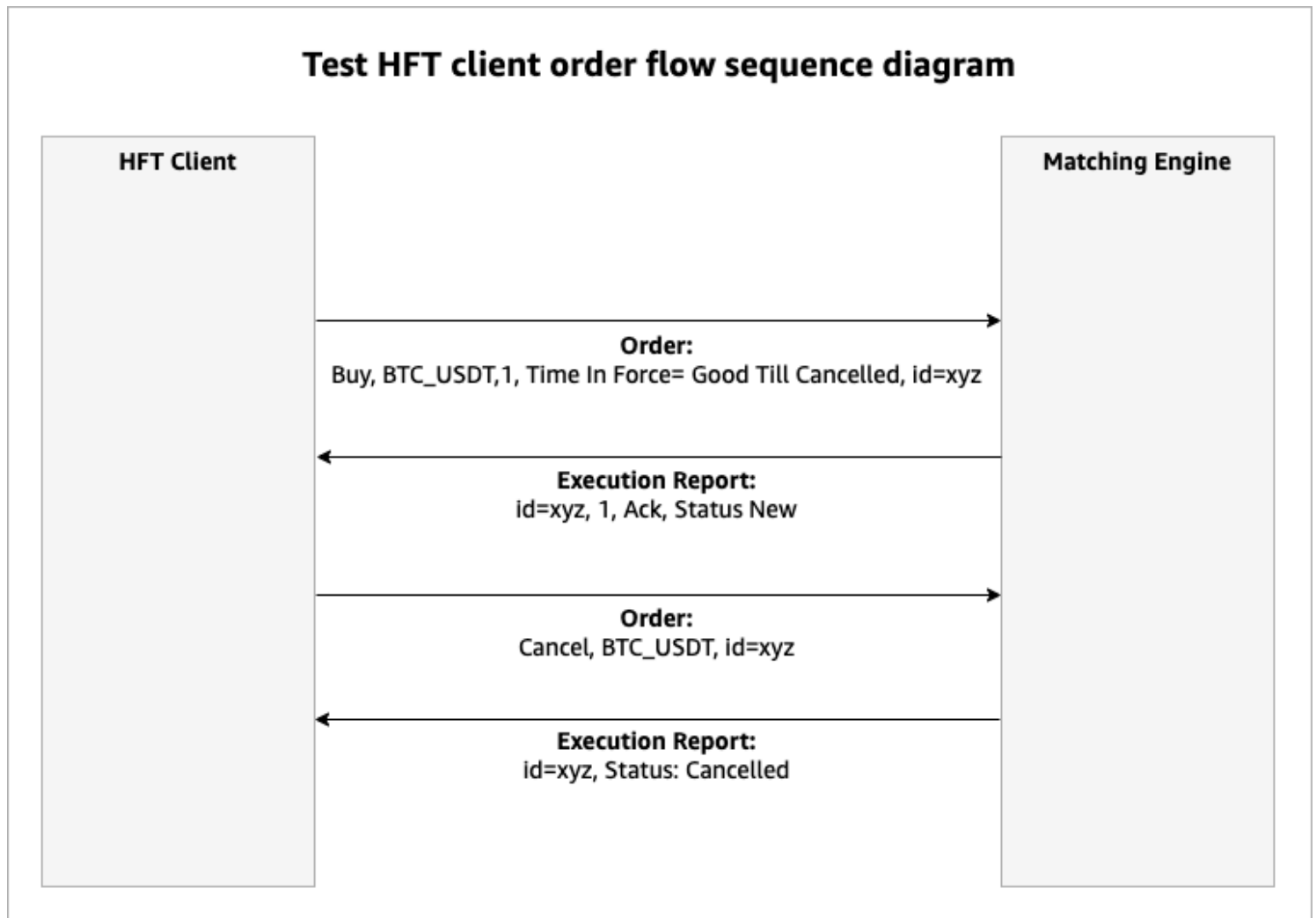


Figure 5. Test HFT client order flow sequence diagram

Once this order cycle is completed, the entire order flow sequence is repeated. Multiple accounts were created on the exchange and this process was executed in parallel. The business logic, therefore, tests both sequential and parallel execution on both the trading engine and exchange sides. Parallel execution also allowed for the generation of different order flow throughputs—low and high rates.

Low-rate configurations generated 10K messages per second and high-rates 400K messages per second, at an average message payload size of 120 bytes.

HFT client optimizations

Testing was conducted on [Amazon EC2 c6id.metal instances](#) running [Amazon Linux 2023](#). The following application layer optimizations and techniques were implemented for the HFT client:

1. **Thread processor affinity through CPU core pinning:** To reduce latency caused by the copying of thread data and instructions into CPU cache, each thread is pinned to a distinct core. The operating system ensures a given thread only executes on a specific core.
2. **Composite buffers:** The HFT client implements [Netty](#) as the underlying application networking framework. Composite buffers in Netty reduce unnecessary object allocations and copy operations when merging multiple frames of data.

3. **IO_uring:** [IO_uring](#) is an asynchronous I/O interface for the Linux kernel. It implements shared memory ring buffers that provide a queue between the application and kernel space, reducing latency by eliminating additional system calls for application I/O operations.
4. **Thread segregation:** Threads responsible for network I/O are kept distinct from those that calculate the round-trip latencies and generate histogram data. This single responsibility model prevents latency incurred from business logic impacting order message transmission.
5. **Reduce pressure from garbage collection (GC):** Various techniques are used including warming up Java virtual machine (JVM) processes to prevent repeated interpretation and compile native code for use from the cache, regular process restarts and specific JVM parameters to reduce GC pressure.

We have made the HFT client available in this [GitHub repository](#), where you can view the code and read more about how these optimizations are implemented.

In the interest of maintaining a straightforward baseline many additional stack optimizations that are typically implemented for specific HFT workload types were not applied for this testing. Workload types not applied were: IRQ handling, CPU P-state and C-state controls, network buffers, kernel bypass, receive side scaling, transmit packet steering, Linux scheduler policies and AWS Elastic Network Adapter tuning.

Testing results

The tests were performed simultaneously across all test network topologies for a continuous 24-hour period. For the purposes of clarity and greatest utility, round-trip times do not include latency added by HFT client business logic and are therefore a clear representation of the network performance cost for each topology. The following table (Figure 6) displays the aggregated results.

| Aggregated round-trip time results for all test network topologies | | | | | |
|--|--------------------|---|--|---|---|
| Test Type | Result Percentiles | Connectivity Tier-1: Amazon VPC Peering with shared Amazon EC2 CPGs | Connectivity Tier-2: Amazon VPC Peering | Connectivity Tier-3: AWS PrivateLink | Connectivity Tier-4: Internet via Amazon CloudFront |
| Low Rate: 10K msgs/sec | P50 | 74 μ s | 133 μ s | 289 μ s | 1.17 ms |
| | P90 | 78 μ s | 144 μ s | 332 μ s | 1.75 ms |
| | P95 | 79 μ s | 147 μ s | 348 μ s | 2.2 ms |
| | P99 | 83 μ s | 153 μ s | 387 μ s | 4.3 ms |
| | P99.9 | 100 μ s | 164 μ s | 510 μ s | 14.3 ms |
| | P99.99 | 120 μ s | 191 μ s | 999 μ s | 47 ms |
| High Rate: 400K msgs/sec | P50 | 96 μ s | 162 μ s | 408 μ s | 1.55 ms |
| | P90 | 107 μ s | 180 μ s | 463 μ s | 2.11 ms |
| | P95 | 112 μ s | 186 μ s | 485 μ s | 2.61 ms |
| | P99 | 124 μ s | 211 μ s | 586 μ s | 5.5 ms |
| | P99.9 | 332 μ s | 461 μ s | 1.46 ms | 15.9 ms |
| | P99.99 | 2.02 ms | 4.9 ms | 6.1 ms | 50 ms |

Figure 6. Aggregated round-trip time results for all test network topologies

The results obtained demonstrate that using Amazon VPC peering and shared Amazon EC2 cluster placement groups, for high message rates at P99, is 98% faster than access by using the Internet with Amazon CloudFront and 41% faster than access by using Amazon VPC peering without the use of shared CPGs.

Conclusion

In this blog we discussed how we worked with One Trading during the pre-launch activities for their new F.A.S.T. exchange product. We demonstrated that our HFT client and the One Trading exchange, deployed on Amazon EC2 c6id.metal instances, are capable of scaling to process a large number of orders per second with minimal application contention. Latency was introduced as a result of the various test network topologies deployed.

We showed how implementing an architecture involving Amazon VPC peering and shared Amazon EC2 cluster placement groups provide the lowest latency connectivity pattern.

While all topologies tested are functionally valid, and can be provided by One Trading to different categories of market participants, Amazon VPC peering and shared CPGs provides a materially beneficial access tier for their largest market making customers.

If you are interested in learning more about One Trading's new F.A.S.T. exchange product please contact [One Trading](#). If you want to run similar performance tests by using or modifying the code created for this HFT client, please contact an [AWS Representative](#).

Further reading

- [Understanding AWS Infrastructure – Regions and Zones](#)
- [Crypto market-making latency and Amazon EC2 shared placement groups](#)
- [Low latency cloud-native exchanges](#)
- [Aeron performance enables capital markets to move to the cloud on AWS](#)

TAGS: [Capital Markets](#)

Comments