Scalable and Fair Multicast for Financial Exchanges in the Cloud

Muhammad Haseeb
New York University
USA

Jinkun Geng
Stanford University
USA

Ulysses Butler
New York University
USA

Xiyu Hao
New York University
USA

Daniel Duclos-Cavalcanti
Technical University of Munich
Germany

Anirudh Sivaraman
New York University
USA

CCS CONCEPTS

• **Networks → Application layer protocols**; **Overlay and other logical network structures**.

KEYWORDS

Cloud, Financial Exchanges, Multicast, Confidential Computing

ACM Reference Format:
Muhammad Haseeb, Jinkun Geng, Ulysses Butler, Xiyu Hao, Daniel Duclos-Cavalcanti, and Anirudh Sivaraman. 2024. Scalable and Fair Multicast for Financial Exchanges in the Cloud. In *ACM SIGCOMM 2024 Conference (ACM SIGCOMM '24), August 4–8, 2024, Sydney, NSW, Australia.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3672202.3673728

There has been a growing interest from industry [9, 13, 14] and academia [8, 10] in migrating financial exchanges to the public cloud because of multiple benefits provided by the cloud, e.g., , robust infrastructure, and potential cost savings [4]. However, migrating financial exchanges from on-prem clusters to the cloud poses several challenges.

Financial Exchanges Requirements: One fundamental requirement for a typical financial exchange is a fair multicast service [8]. Such a service (e.g., NASDAQ's ITCH [12]) is responsible for disseminating information about the state of the market (termed as *market data*) to a large number of market participants (MPs). Market data serves as an important reference for MPs to make trading decisions, and High Frequency Trading (HFT) firms often compete on how quickly they can place trades based on this information. To ensure *fairness*, market data should be delivered to every MP almost simultaneously so that no MP can earn an unfair advantage over another MP. Besides, HFT firms also require consistently *low latency* from exchange to MPs for market data distribution so that MPs can trade on the most up-to-date information.

Challenges in the Cloud: While a high-performance fair multicast service in on-prem clusters might be implemented using switch support and carefully engineered networks,[1] the situation is much less favorable in the public cloud, where the hardware (e.g., switch) support for multicast is not usually available to cloud tenants. The public cloud also exhibits higher and more varied latency than on-prem clusters. This variance, combined with the limited control a tenant possesses on the underlying network, makes it difficult to realize fair/simultaneous delivery of market data to all market participants (MPs). As a result, implementing such a multicast service for financial exchanges in the public cloud becomes challenging for cloud tenants.

Jasper: We develop Jasper, an overlay multicast service for cloud-hosted financial exchanges (Figure 1). To achieve low latency while scaling to a large number of receivers, Jasper (1) builds a *tree of proxies* for multicast, instead of having the sender directly unicast its message to all receivers; (2) introduces a *VM hedging* technique to tackle high latency variance where replicated messages traverse through redundant paths; (3) designs a deployment model assuming no trust between MPs and the exchange server that incorporates a *hold-and-release* mechanism [8] (the receivers hold messages and only process them at a deadline) with the *Trusted Execution Environment* (TEE) technique [2] to ensure fairness.

A Proxy Tree: A tree reduces the serialization or transmission delay required to unicast multiple messages back-to-back, but adds additional hops in the sender-to-receiver path. In Jasper, we use the tree structure as a starting point and develop new techniques to lower latency and latency variance and achieve fairness in data delivery while scaling to a large number of receivers. The structure of a proxy[2] tree (depth D and fan-out F) is tuned to minimize latency. Existing cloud-based exchanges [8–10] implement multicast using the direct unicast approach. This may be considered a special case of a tree where D is 1 and F is N. We show that there is value in increasing the D and decreasing the F as the number of receivers (N) grows. We provide a heuristic for tuning D and F which provides good performance: Given N receivers, we fix $F = 10$ and then derive $D = [log_{10}N]$ (round to the nearest integer). We find that more sophisticated learning-based techniques do not outperform our simple heuristic because of the high variation in latency and performance of VMs in the public cloud.

VM hedging: For achieving consistently low latency and decreasing the latency variance spatially, Jasper introduces a technique called VM hedging, motivated by request hedging [6, 15, 17]. In VM hedging, each VM in the proxy tree receives messages from two or more different sources (i.e., parent and one or more aunts), defined by hedging factor H, where the path lengths for all messages are the same. A VM processes and forwards the message copy to

[1] Some financial firms equalize wire lengths [16] to achieve simultaneous delivery of market data to all MPs.

[2] The proxies are separate VMs. Traders' VMs do not act as proxies/intermediaries.

ACM SIGCOMM '24, August 4–8, 2024, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0717-9/24/08…$15.00
https://doi.org/10.1145/3672202.3673728

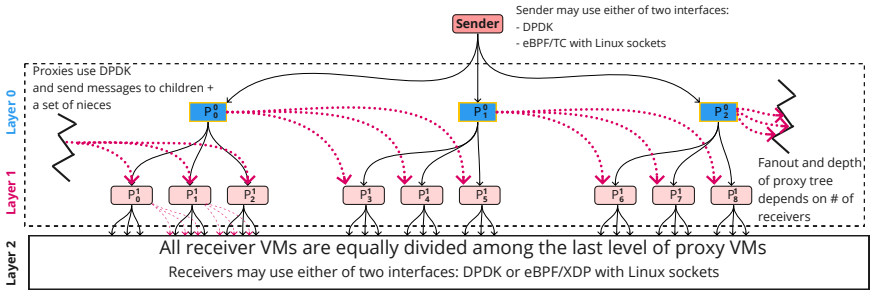

Fig. 1: A Jasper Tree. Dotted edges represent hedging, which lowers the latency variance.

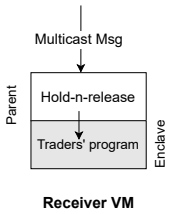

Fig. 2: A receiver VM is partitioned into a parent VM and a secure enclave (e.g., [2])

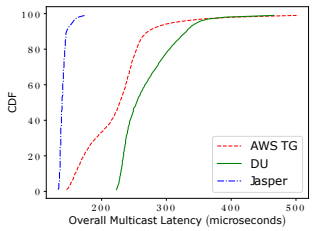

Fig. 3: Jasper outperforms DU, and AWS.

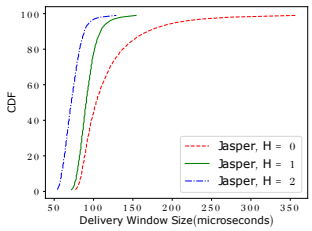

Fig. 4: Hedging reduces delivery window.

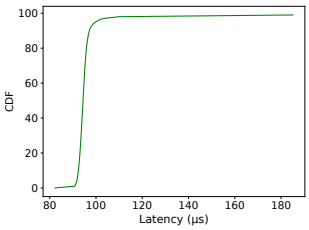

Fig. 5: Enclave has high latency.

children that is received earliest and discards the rest. VM hedging reduces the impact of latency fluctuations, yields much smaller latency variance, and narrows down the window of time in which all the multicast receivers receive a multicast message.

Jasper Deployment Model: To ensure perfect fair delivery of data, the exchange requires the MPs to run a hold-and-release mechanism, as proposed by CloudEx [8]. In this mechanism, deadlines are attached to the messages, and an MP is supposed to receive a message and only process it at or after a deadline associated with the message. The deadlines are set, by the exchange in a way to ensure that every MP will receive the message by the deadline. The efficacy of this fair delivery mechanism relies on the MPs to respect the hold-and-release protocol. However, MPs have the incentive to not respect the protocol and process the messages before the deadline to gain an advantage over the others, while they also do not want the reveal their program to the exchange. To resolve this dilemma, we incorporate TEEs, which have become generally available in public cloud [2, 3, 5], to maintain security boundaries between the exchange and traders. More specifically, MPs run their trading algorithm inside the enclaves of the VMs owned by the exchange, and the exchange runs the hold-and-release mechanism outside the enclave (see Fig. 2). As a result, the MPs' trading program is protected by the TEE, and the exchange can execute the hold-and-release protocol in its owned VMs. We benchmark the latency of a TEE offered by AWS called Nitro Enclave. Figure 5 shows a high latency associated with it. In the future, we will work to improve this latency. Other results in this article, do not use TEE.

Jasper outperforms AWS Transit Gateway: Figure 3 shows that Jasper distinctly outperforms AWS-TG based multicast and DU (Direct Unicasts approach). The median latency for Jasper is 129 µs. whereas it is 228 µs for AWS TG and 254 µs for DU for 100 multicast receivers. We observe more benefits from Jasper as the number of receivers increases. We synchronize the clocks of using [7] with an O(100) ns accuracy. The benchmark does not use TEE, which would add an almost constant overhead on every technique.

VM Hedging reduces delivery window size: Figure 4 shows that delivery window size (the difference between the time when the first receiver and the last receiver receive a multicast message) is reduced with hedging. Note that lost packets are ignored which is a common practice in the on-prem exchanges as any recovered packet would already be too late to be useful by the MPs. In our preliminary benchmarks, we witness a significantly low loss rate.

Next Steps: Currently Jasper mainly focuses on performance improvement in the outbound direction (i.e., market data dissemination from the exchange to MPs). However, the cloud environment also presents challenges in the inbound direction (i.e., order submission from a large number of MPs to an exchange server). We have several avenues to explore as the next steps:

(1) A **distributed limit order book** [1] would enable the exchange to scale horizontally and improve the throughput of the system.

(2) A **bucketed integer priority queue**, instead of simple priority queues at the exchange, is desirable to build high-performance exchange systems, which have lower time complexity and may enhance the throughput of the exchange.

(3) **Smartly prioritizing the critical orders** (i.e., orders that are more likely to get executed) when orders are sent from MPs to the exchange server through the proxy tree. Such prioritized order submission would enhance market liquidity by matching more orders per unit of time.

A detailed technical report of this project is available [11].

REFERENCES

[1] Frédéric Abergel, Marouane Anane, Anirban Chakraborti, Aymen Jedidi, and Ioane Muni Toke. 2016. *Limit Order Books.* Cambridge University Press.

[2] AWS. 2024. AWS Nitro Enclaves. https://aws.amazon.com/ec2/nitro/nitro-enclaves/. Accessed: 2024-05-22.

[3] Azure. 2023. Trusted Execution Environment (TEE). https://learn.microsoft.com/en-us/azure/confidential-computing/trusted-execution-environment. Accessed: 2024-05-22.

[4] Sara Castellanos. [n. d.]. Nasdaq Ramps Up Cloud Move. https://www.wsj.com/articles/nasdaq-ramps-up-cloud-move-11600206624. Accessed: 2024-01-31.

[5] Alibaba Cloud. 2024. Build an SGX confidential computing environment. https://www.alibabacloud.com/help/en/ecs/user-guide/build-an-sgx-encrypted-computing-environment. Accessed: 2024-05-22.

[6] Jeffrey Dean and Luiz André Barroso. 2013. The Tail at Scale. *Commun. ACM* 56 (2013), 74–80. http://cacm.acm.org/magazines/2013/2/160173-the-tail-at-scale/fulltext

[7] Yilong Geng, Shiyu Liu, Zi Yin, Ashish Naik, Balaji Prabhakar, Mendel Rosenblum, and Amin Vahdat. 2018. Exploiting a Natural Network Effect for Scalable, Fine-grained Clock Synchronization. In *Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation* (Renton, WA, USA) *(NSDI'18)*. USENIX Association, Berkeley, CA, USA, 81–94.

[8] Ahmad Ghalayini, Jinkun Geng, Vighnesh Sachidananda, Vinay Sriram, Yilong Geng, Balaji Prabhakar, Mendel Rosenblum, and Anirudh Sivaraman. 2021. CloudEx: A Fair-Access Financial Exchange in the Cloud. In *Proceedings of the Workshop on Hot Topics in Operating Systems* (Ann Arbor, Michigan) *(HotOS '21)*. Association for Computing Machinery, New York, NY, USA, 96–103. https://doi.org/10.1145/3458336.3465278

[9] Junzhi Gong, Yuliang Li, Devdeep Ray, KK Yap, and Nandita Dukkipati. 2024. Octopus: A Fair Packet Delivery Service. *arXiv preprint arXiv:2401.08126* (2024).

[10] Eashan Gupta, Prateesh Goyal, Ilias Marinos, Chenxingyu Zhao, Radhika Mittal, and Ranveer Chandra. 2023. DBO: Fairness for Cloud-Hosted Financial Exchanges. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 550–563. https://doi.org/10.1145/3603269.3604871

[11] Muhammad Haseeb, Jinkun Geng, Ulysses Butler, Xiyu Hao, Daniel Duclos-Cavalcanti, and Anirudh Sivaraman. 2024. Jasper: Scalable and Fair Multicast for Financial Exchanges in the Cloud. arXiv:2402.09527

[12] NASDAQ. [n. d.]. Nasdaq TotalView-ITCH 5.0. https://www.nasdaqtrader.com/content/technicalsupport/specifications/dataproducts/NQTVITCHSpecification.pdf. Accessed: 2024-02-02.

[13] Nasdaq.com. [n. d.]. Nasdaq and AWS Partner to Transform Capital Markets. https://www.nasdaq.com/press-release/nasdaq-and-aws-partner-to-transform-capital-markets-2021-12-01. Accessed: 2024-01-26.

[14] Alexander Osipovich. [n. d.]. Google Invests 1 Billion in Exchange Giant CME, Strikes Cloud Deal. https://www.wsj.com/articles/google-invests-1-billion-in-exchange-giant-cme-strikes-cloud-deal-11636029900. Accessed: 2021-02-02.

[15] Mia Primorac, Katerina Argyraki, and Edouard Bugnion. 2021. When to Hedge in Interactive Services. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, 373–387. https://www.usenix.org/conference/nsdi21/presentation/primorac

[16] Andrew Smith. 2014. Fast Money: The Battle Against the High Frequency Traders. *The Guardian* 7 (2014).

[17] Ashish Vulimiri, Philip Brighten Godfrey, Radhika Mittal, Justine Sherry, Sylvia Ratnasamy, and Scott Shenker. 2013. Low latency via Redundancy. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies* (Santa Barbara, California, USA) *(CoNEXT '13)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2535372.2535392