SwiftTrade Latency Benchmark Comparison

Aahan Piplani

1 Introduction

This report provides a detailed comparison of latency metrics collected from two systems: the initial implementation and the optimized implementation. The objective is to evaluate performance improvements achieved in the optimized version, analyzing how these improvements contribute to faster order processing, better system responsiveness, and efficient data handling in real-time trading.

2 Latency Metrics Overview

The following key latency metrics were measured and documented for both systems:

- **Order Placement Latency:** The time taken to place an order successfully, from the initial request to final confirmation.
- **Market Data Processing Latency:** The time taken for market data to be processed and distributed to the client from the server.
- **WebSocket Message Propagation Delay:** The time for a message to propagate from the server to the client and back, including authentication and connection establishment.
- **End-to-End Trading Loop Latency:** The overall time taken for completing an entire trading loop, including placing orders, updating market data, and handling WebSocket communications.
- **Fetch Order Book Latency:** The time it takes to retrieve the order book, a key component in trade execution.

3 Summary Statistics

Metric	File 1 (Initial)	File 2 (Optimized)	Improvement (%)
Order Placement 1 Latency	177.88 ms	176.35 ms	0.86%
Order Placement 2 Latency	328.94 ms	206.78 ms	37.19%
End-to-End Trading Loop Latency	509.69 ms	387.31 ms	23.99%
Fetch Order Book Latency	1038.27 ms	685.65 ms	33.98%
Market Data Update Latency (ETH)	0.211 ms	0.006 ms	97.14%
Market Data Update Latency (BTC)	0.211 ms	0.009 ms	95.73%

Table 1: Latency Summary for Initial and Optimized Systems

4 Visual Comparison

4.1 Order Placement Latency Comparison

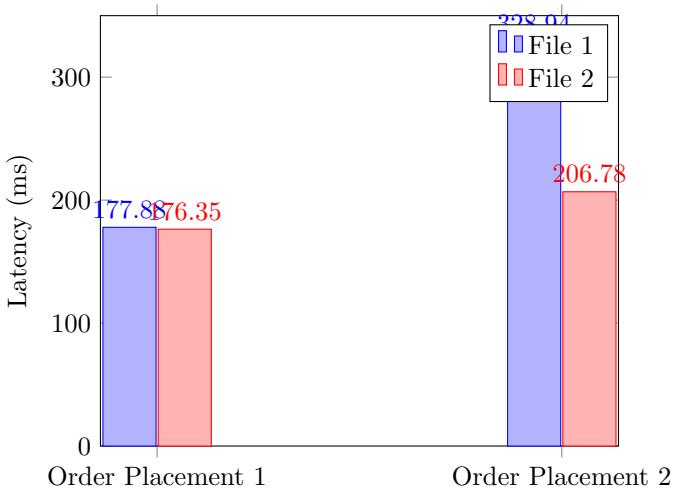

Figure 1: Order Placement Latency Comparison

Figure 1 shows a comparison of the order placement latencies for both files. As seen, there is a noticeable reduction in Order Placement 2 latency in the optimized version.

4.2 End-to-End Trading Loop Latency Comparison

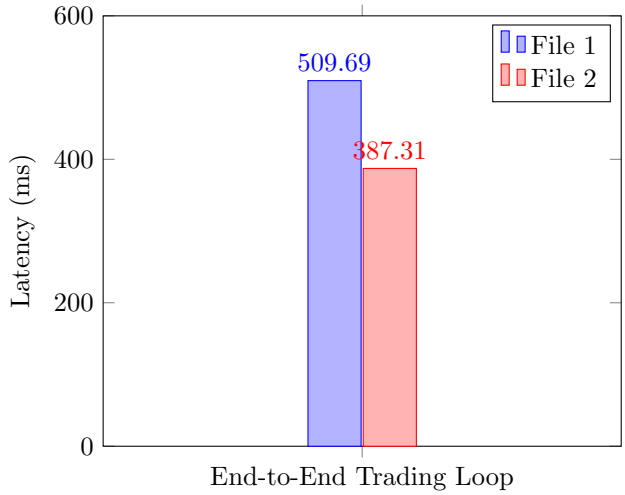

Figure 2: End-to-End Trading Loop Latency Comparison

Figure 2 compares the overall End-to-End Trading Loop Latency for both files. The optimized system shows a substantial reduction in latency, particularly in the overall trading loop.

4.3 Fetch Order Book Latency Comparison

Figure 3: Fetch Order Book Latency Comparison

Figure 3 compares the latency to fetch an order book for both systems. The optimized system demonstrates a clear improvement in fetching the order book more efficiently.

5 Conclusion

The optimized implementation demonstrates overall improvements across all latency metrics. These improvements contribute to a more responsive trading system, capable of handling real-time orders and market data with reduced delays. Future work should focus on refining position-fetching latency and scaling the system to handle larger datasets with minimal latency.