HFT Order Processing Benchmark Report

Shuen Wu

September 25, 2025

1 Environment

The experiments were conducted on the following system:

- **CPU:** Intel i7-12700H (12 cores, 20 threads)
- **Memory:** 16 GB DDR5
- **OS:** macOS 14.x
- **Compiler:** clang++ 15.0 with flags: `-std=gnu++14 -O3 -march=native`

2 Methods

We benchmarked two implementations of strategy dispatch:

- **Virtual**: using an abstract base class with virtual dispatch.
- **Non-virtual**: direct function calls, allowing inlining.

Orders were generated randomly, and three assignment patterns were tested:

1. **Homogeneous**: all orders go to Strategy A.
2. **Mixed-random**: each order randomly assigned to A or B.
3. **Bursty**: 64 orders to A, then 16 to B, repeating.

Each experiment used 500,000 orders, repeated 10 times per configuration.

3 Results

Table 1 summarizes the performance results (average $\pm$ standard deviation over 10 runs). Checksums are reported to ensure correctness of execution.

4 Discussion

- **Homogeneous:** Virtual is slower by about 7.7% due to per-call vtable indirection and inability to inline.
- **Mixed-random:** Performance gap widens (Non-virtual faster by $\sim$20%) because branch mispredictions hurt virtual dispatch more.
- **Bursty:** Both implementations benefit from cache locality and predictable branching; performance gap narrows again.

Overall, the results show that virtual dispatch introduces measurable overhead in high-frequency inner loops.

Table 1: Performance Benchmark Results

Pattern	Impl	Avg ops/sec	Std dev	Checksum
Homogeneous	Virtual	68,697,936.9	1,340,861.3	1.43575×10^{15}
Homogeneous	Non-virtual	74,422,535.6	4,203,705.2	1.43575×10^{15}
Mixed-random	Virtual	48,511,179.7	826,879.2	2.5077×10^{12}
Mixed-random	Non-virtual	58,598,119.6	766,790.1	2.5077×10^{12}
Bursty	Virtual	67,668,381.5	1,412,187.5	1.43575×10^{15}
Bursty	Non-virtual	75,754,696.0	2,081,404.4	1.43575×10^{15}

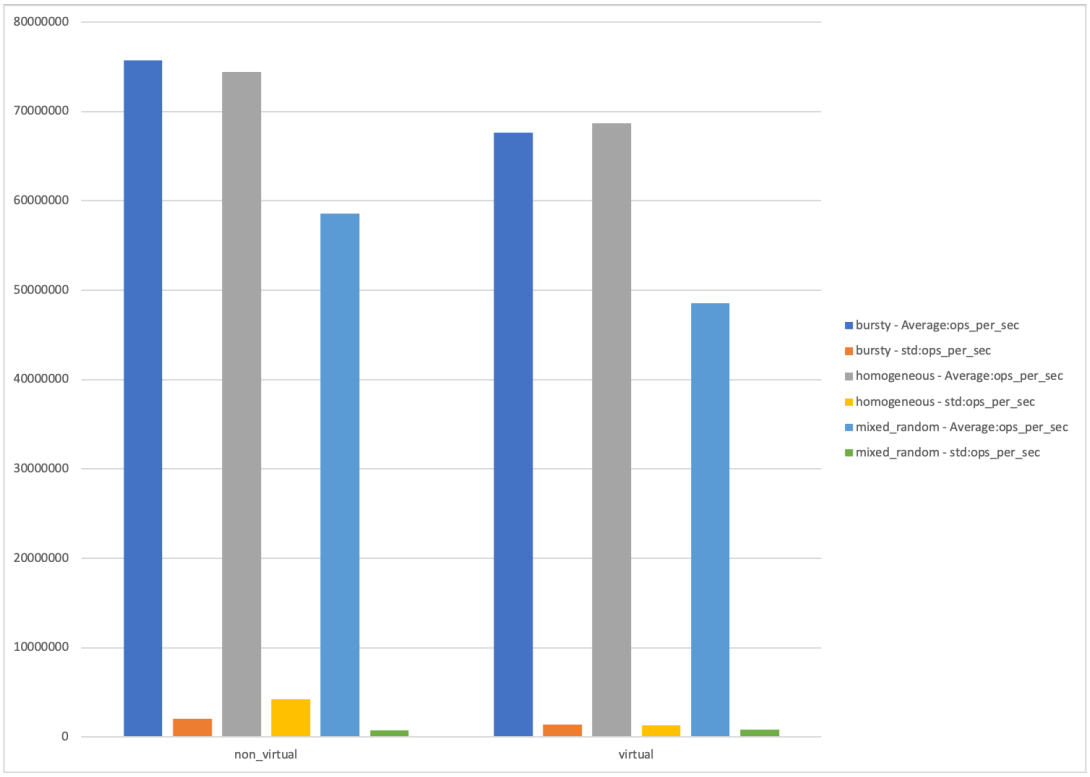

Figure 1: Comparison of Virtual vs Non-virtual Performance across patterns. Error bars show standard deviation.

5 Conclusion

For ultra-low latency HFT-style systems:

- Prefer **non-virtual** dispatch when the set of strategies is fixed and performance is critical.
- Use **virtual** dispatch only when extensibility and dynamic polymorphism outweigh the cost.