Low latency stock exchange

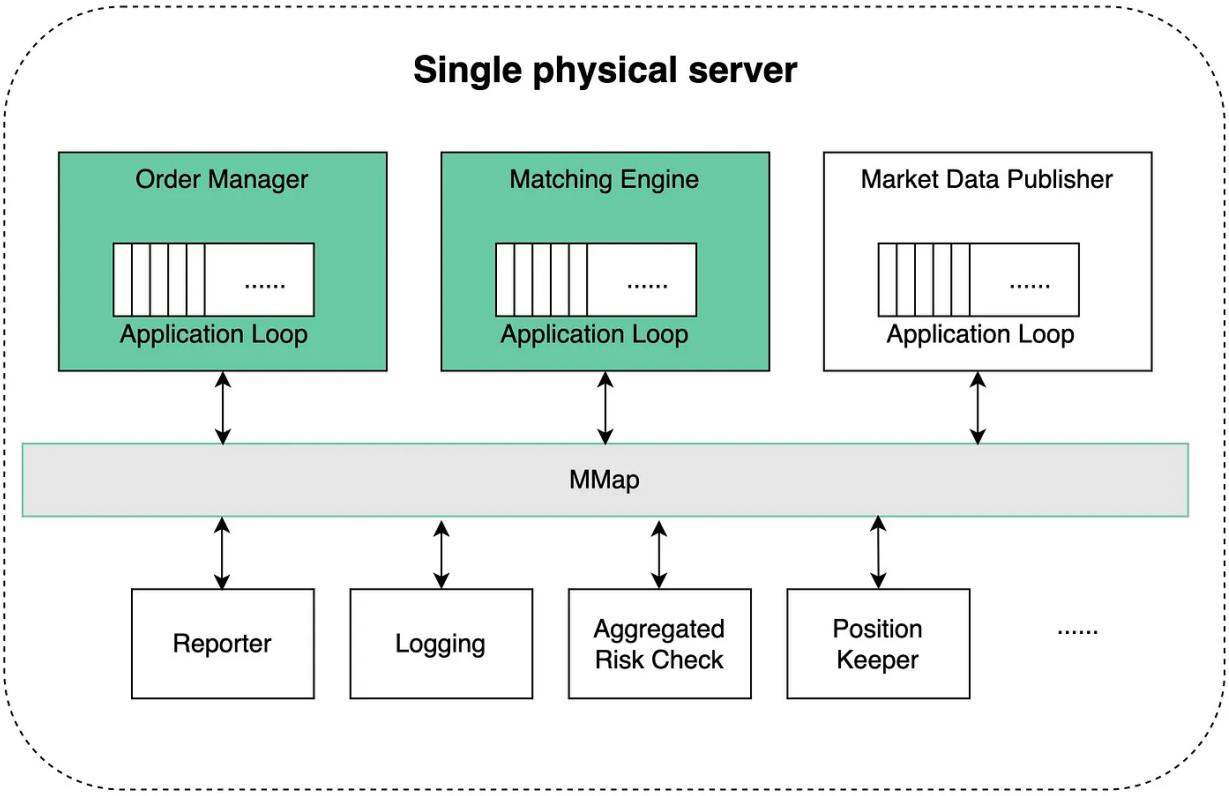
ALEX XU
DEC 10, 2021

How does a modern stock exchange achieve **microsecond latency**? The principal is:

Do less on the critical path
- Fewer tasks on the critical path
- Less time on each task
- Fewer network hops
- Less disk usage

For the stock exchange, the critical path is:
- **start**: an order comes into the order manager
- mandatory risk checks
- the order gets matched and the execution is sent back
- **end**: the execution comes out of the order manager

Low Latency Stock Exchange Design



Other non-critical tasks should be removed from the critical path.

We put together a design as shown in the diagram:

- deploy all the components in a single giant server (no containers)

- use shared memory as an event bus to communicate among the components, no hard disk

- key components like Order Manager and Matching Engine are single-threaded on the critical path, and each pinned to a CPU so that there is **no context switch** and **no locks**

- the single-threaded application loop executes tasks one by one in sequence

- other components listen on the event bus and react accordingly

If you enjoyed those posts, you might like our system design interview books as well.

SDI-vol1: https://amzn.to/3tK0qQn

SDI-vol2: https://amzn.to/37ZisW9

55 Likes

3 Comments

Write a comment...

Andre Nov 7, 2022

Do you have any suggestions for this memory event bus?

LIKE REPLY SHARE

Edwin Torres Sep 9, 2022

I don't understand very well why do we need a single physical server (I think I got the point that networking adds latency), but what about scalability?

LIKE REPLY SHARE

1 reply

1 more comment...