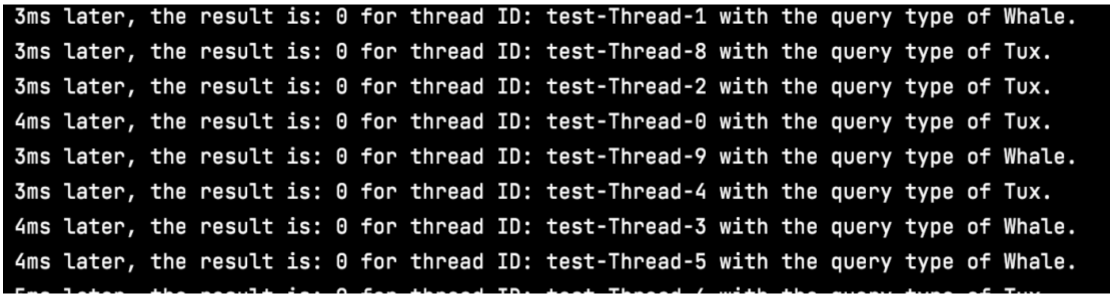
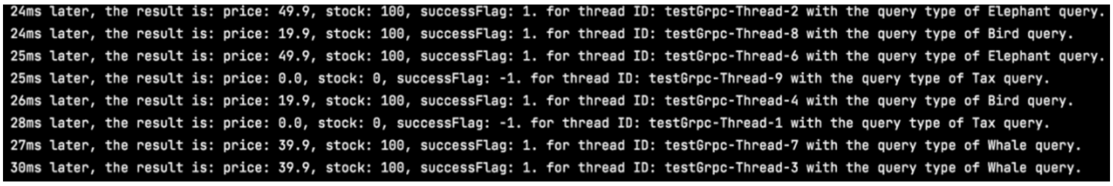
Measurements

Measure 1

Firstly, we do the comparison between latency of query between part 1 and part 2, each of both has concurrency of 10 threads:



```
3ms later, the result is: 0 for thread ID: test-Thread-1 with the query type of Whale.
3ms later, the result is: 0 for thread ID: test-Thread-8 with the query type of Tux.
3ms later, the result is: 0 for thread ID: test-Thread-2 with the query type of Tux.
4ms later, the result is: 0 for thread ID: test-Thread-0 with the query type of Tux.
3ms later, the result is: 0 for thread ID: test-Thread-9 with the query type of Whale.
3ms later, the result is: 0 for thread ID: test-Thread-4 with the query type of Tux.
4ms later, the result is: 0 for thread ID: test-Thread-3 with the query type of Whale.
4ms later, the result is: 0 for thread ID: test-Thread-5 with the query type of Whale.
```



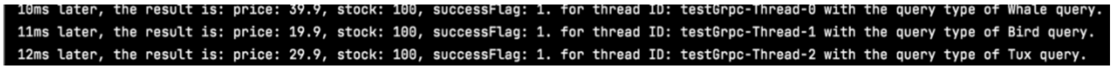
```
24ms later, the result is: price: 49.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-2 with the query type of Elephant query.
24ms later, the result is: price: 19.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-8 with the query type of Bird query.
25ms later, the result is: price: 49.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-6 with the query type of Elephant query.
25ms later, the result is: price: 0.0, stock: 0, successFlag: -1. for thread ID: testGrpc-Thread-9 with the query type of Tax query.
26ms later, the result is: price: 19.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-4 with the query type of Bird query.
28ms later, the result is: price: 0.0, stock: 0, successFlag: -1. for thread ID: testGrpc-Thread-1 with the query type of Tax query.
27ms later, the result is: price: 39.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-7 with the query type of Whale query.
30ms later, the result is: price: 39.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-3 with the query type of Whale query.
```

Here we only plot part of them. We can easily observe that the query time for part 1 is about 3ms, while that for part 2 is about 24ms. It is easy to understand because the gRPC method have more complex mechanism to guarantee transferring between platforms, so naturally it goes slower.

Measure 2

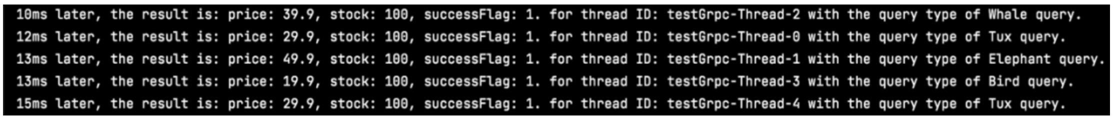
For part 2, we have tested the situations of 3, 5, 10, 100 clients, measuring their latency for buy and query.

3:



```
10ms later, the result is: price: 39.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-0 with the query type of Whale query.
11ms later, the result is: price: 19.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-1 with the query type of Bird query.
12ms later, the result is: price: 29.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-2 with the query type of Tux query.
```

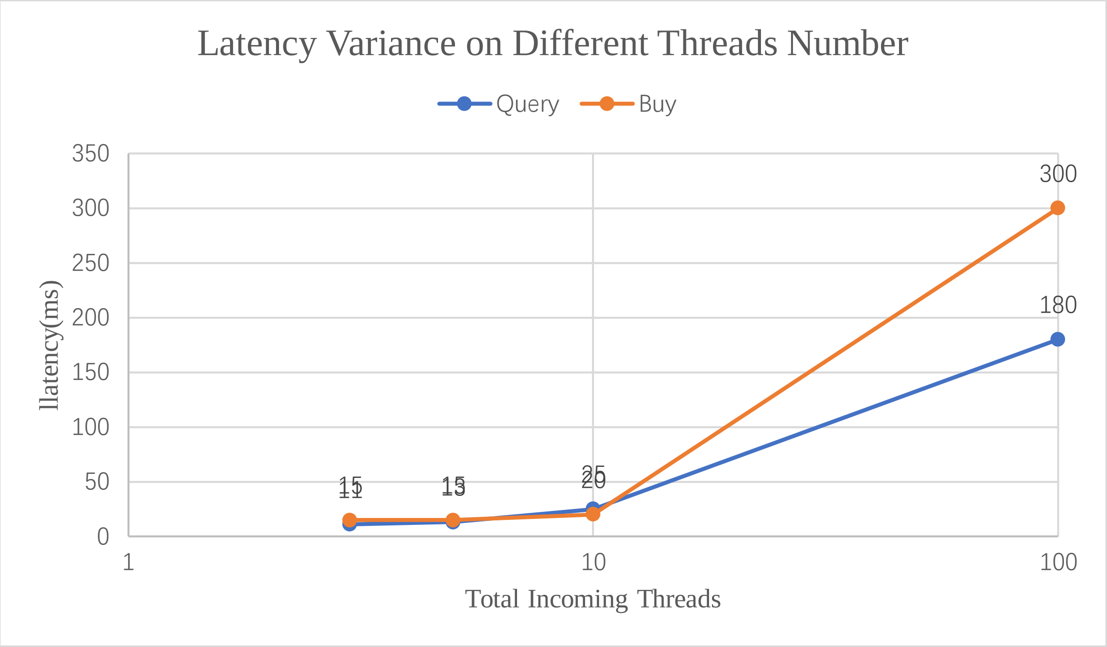
5:



```
10ms later, the result is: price: 39.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-2 with the query type of Whale query.
12ms later, the result is: price: 29.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-0 with the query type of Tux query.
13ms later, the result is: price: 49.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-1 with the query type of Elephant query.
13ms later, the result is: price: 19.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-3 with the query type of Bird query.
15ms later, the result is: price: 29.9, stock: 100, successFlag: 1. for thread ID: testGrpc-Thread-4 with the query type of Tux query.
```

…

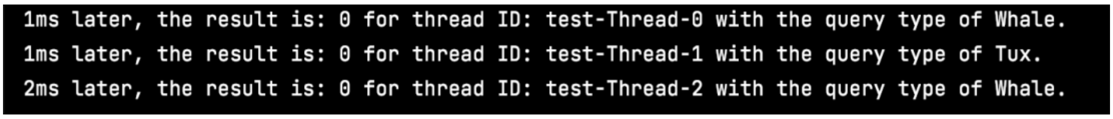
The overall graph is shown below:



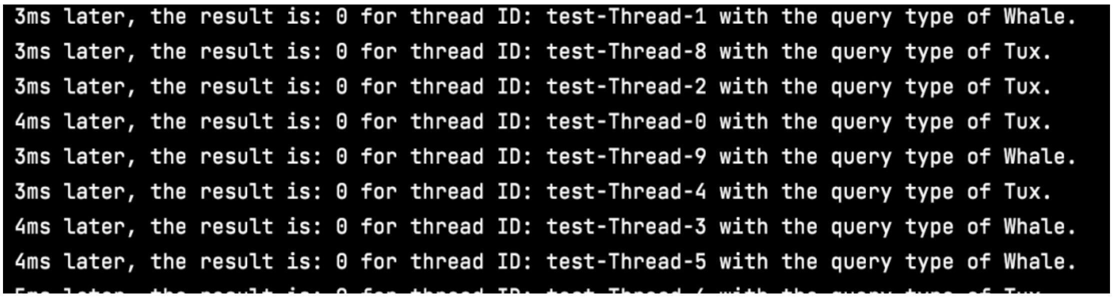
We can see that under high concurrency, buy can be slower because it will write to the stock, and write needs more synchronization than read. Besides, we can see that when the threads/client number increases, the latency goes high which is obvious.

Measure 3

Here we have tested the 3 and 10 threading requests' latency difference towards the part 1 server with the thread pool size of 3.



```
1ms later, the result is: 0 for thread ID: test-Thread-0 with the query type of Whale.
1ms later, the result is: 0 for thread ID: test-Thread-1 with the query type of Tux.
2ms later, the result is: 0 for thread ID: test-Thread-2 with the query type of Whale.
```



```
3ms later, the result is: 0 for thread ID: test-Thread-1 with the query type of Whale.
3ms later, the result is: 0 for thread ID: test-Thread-8 with the query type of Tux.
3ms later, the result is: 0 for thread ID: test-Thread-2 with the query type of Tux.
4ms later, the result is: 0 for thread ID: test-Thread-0 with the query type of Tux.
3ms later, the result is: 0 for thread ID: test-Thread-9 with the query type of Whale.
3ms later, the result is: 0 for thread ID: test-Thread-4 with the query type of Tux.
4ms later, the result is: 0 for thread ID: test-Thread-3 with the query type of Whale.
4ms later, the result is: 0 for thread ID: test-Thread-5 with the query type of Whale.
```

Result turns out to show that when the number of clients is larger than size of static thread pool, the response time increases to about double time.

From the overall 3 measures, we get a sense of how thread pool size, number of incoming clients, read or write operation and different implementations of currency act on the time of server with thread pool to process them.