# 1 Introduction

This report is mainly about how to measure server performance, as well as analyse the scalability of the server.

# 2 Test Method

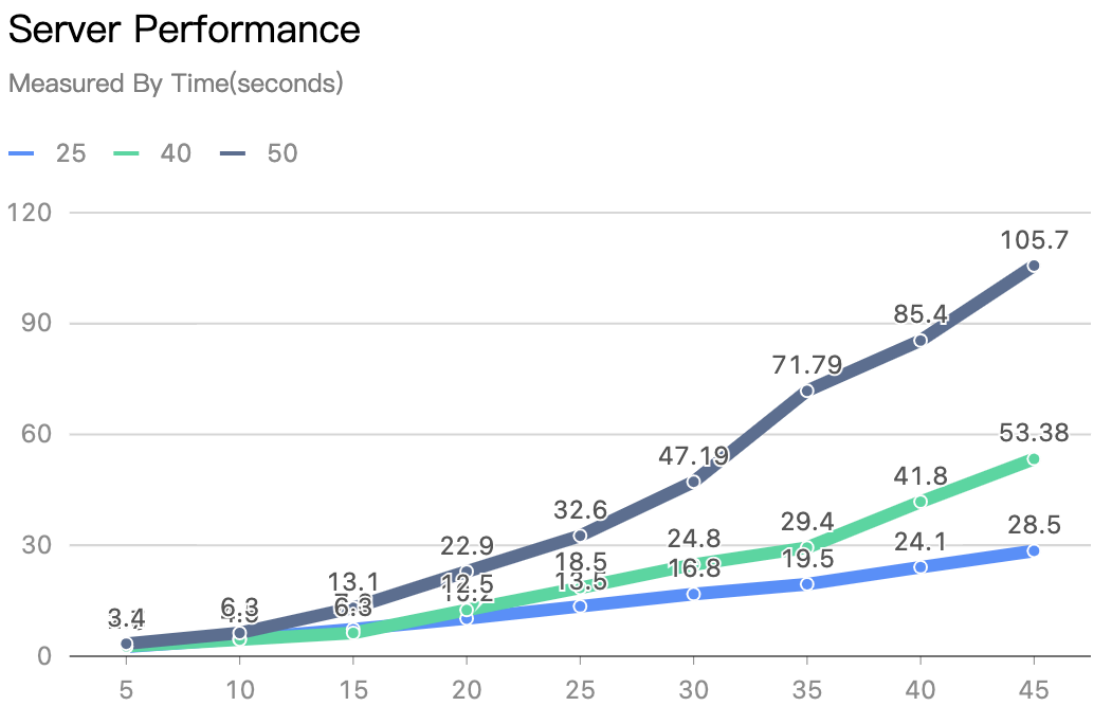
Test Script Location: `erss-hm4-xz295-wg83/src/test/java/demo/IntegrationTest.sh`

**Key Parameters:**

- `ACCOUNT_LEN`: indicate how long the account can be. This is used to test for correctness, preventing for duplicate generation of account numbers
- `WORK_LOAD`: indicate how many clients can send the command at the same time
- `REQUEST_NUM`: how many requests each client can send at one time

## 2.1 Workload Test

- In this section, we have set 25, 40, 50 messages in one client command
- And the x-axis is how many clients are concurrently send the request, which can measure how servers handle this under many clients.
- y-axis is how long it takes for server to finish the request




> **Result**

- We can see as the message increasing per request, the time for server to deal will increase
- When client number increases, for same request, it will take server sometimes 2 times to process.

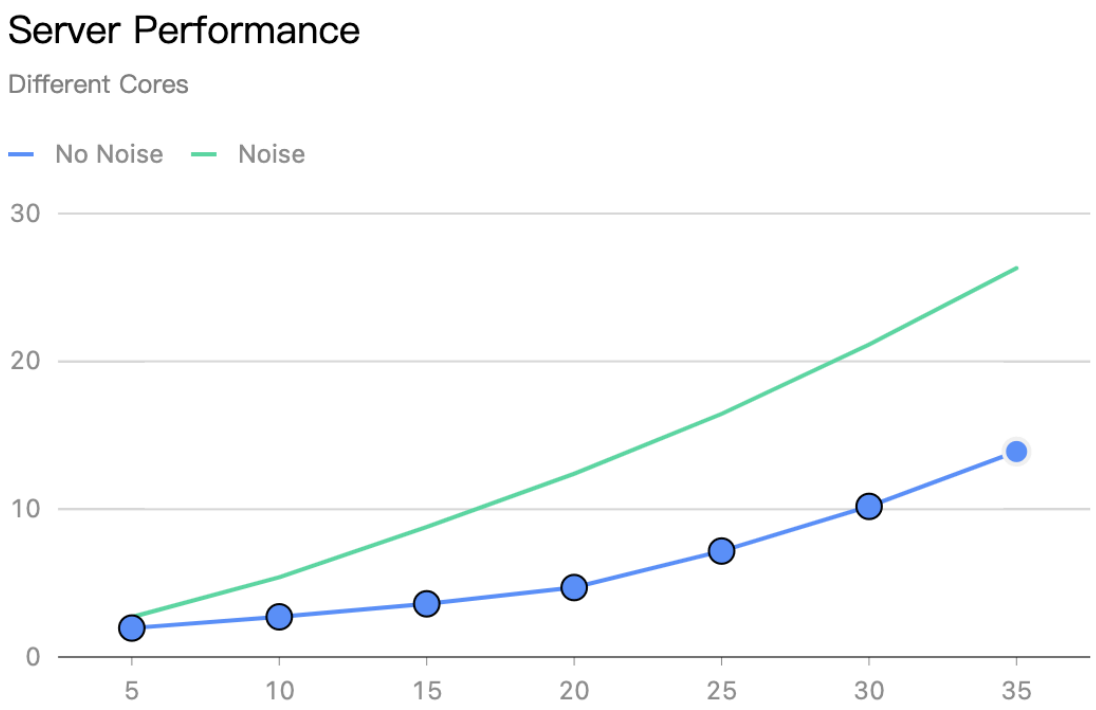
# 2.2 Workload Test With Bad Request Noise

We can forsee once the service is online, there might be some clients sending bad request.

Thus we test under such nodes, if server lowers its performance

> **Method**

- We have launced a bad request thread, which every 0.5 seconds will issue 5X clients to send bad requests



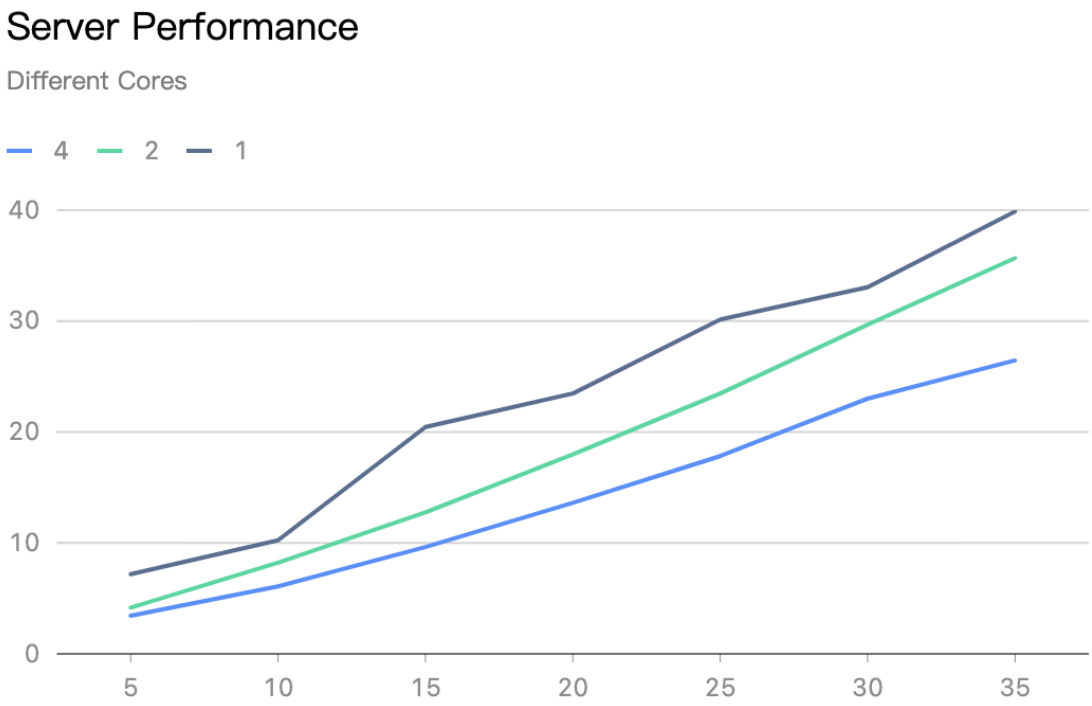

> **Result**

- When being messes with the bad requests, the server has longer time to process that.
- In the future, we can add Proxy-Like protection. Once we find out some of clients who always send us the bad requests, we make it forbidden to visit the service

# 2.3 Different Cores Test

The basic setup is

- under 4, 2, 1 cores, how server performs
  - command `sudo taskset -p 0xf pid`
- the x-axis mean how many clients send the request concurrently
- the y-axis mean how long it takes for server to process that




> **Result**

- We can see as the number of cores increase, the time it spends will be shorter
- but it won't be twice, or four times shorter compared to one core.
    - the main reason could be contention increasing and it takes longer to acquire the lock