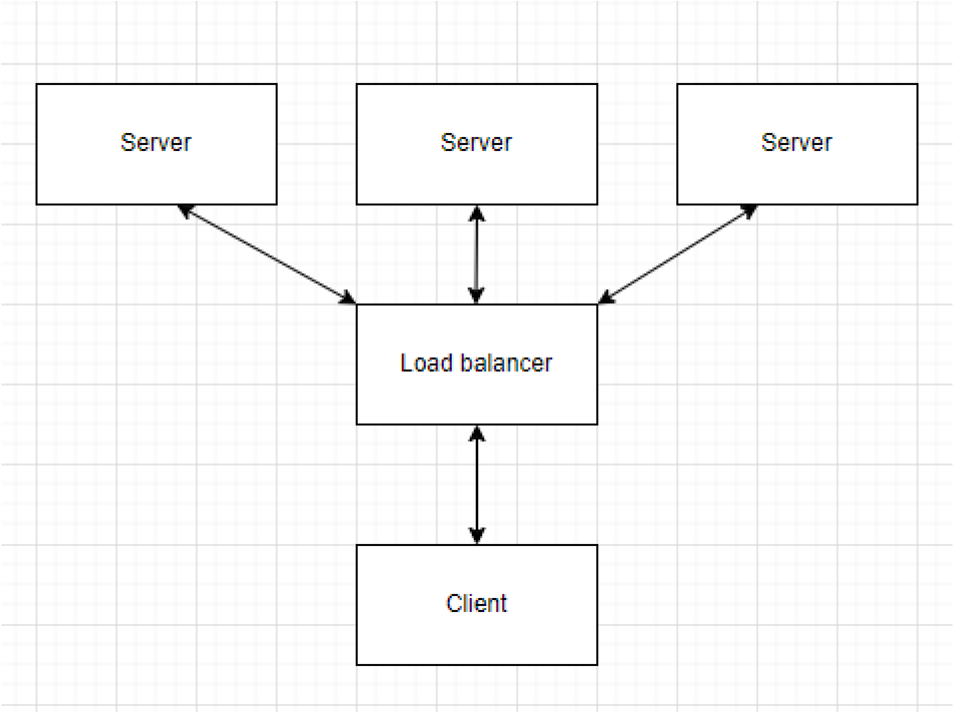
Distributed project

Janne Uutela

Architecture

Servers and loadbalancer are docker containers.

- Python and flask
- For client postman/thunder client and script.



Communication

- Servers send their url to loadbalancer when starting
  - “server_up” endpoint in loadbalancer
  - It signals the loadbalancer that it can start sending traffic.

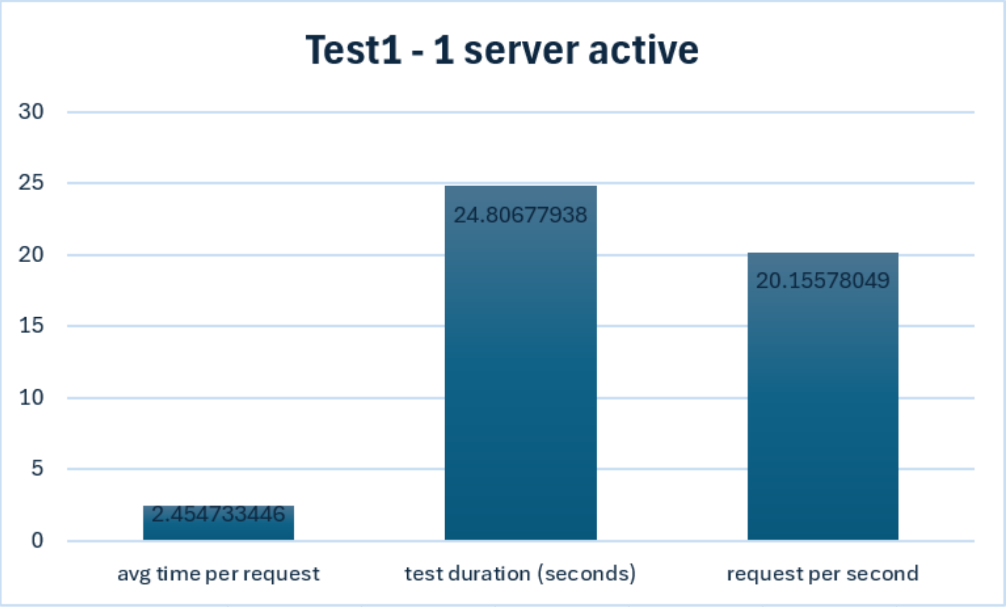
- Loadbalancer divides traffic among the received server urls.
  - Keeps track of (server url, timestamp of last request, and if the server is active)
  - Healthcheck to server endpoint “health_check”
    - When requests are not being currently made
  - When request fails, server is set to not active and does not receive traffic until health check succeeds again.

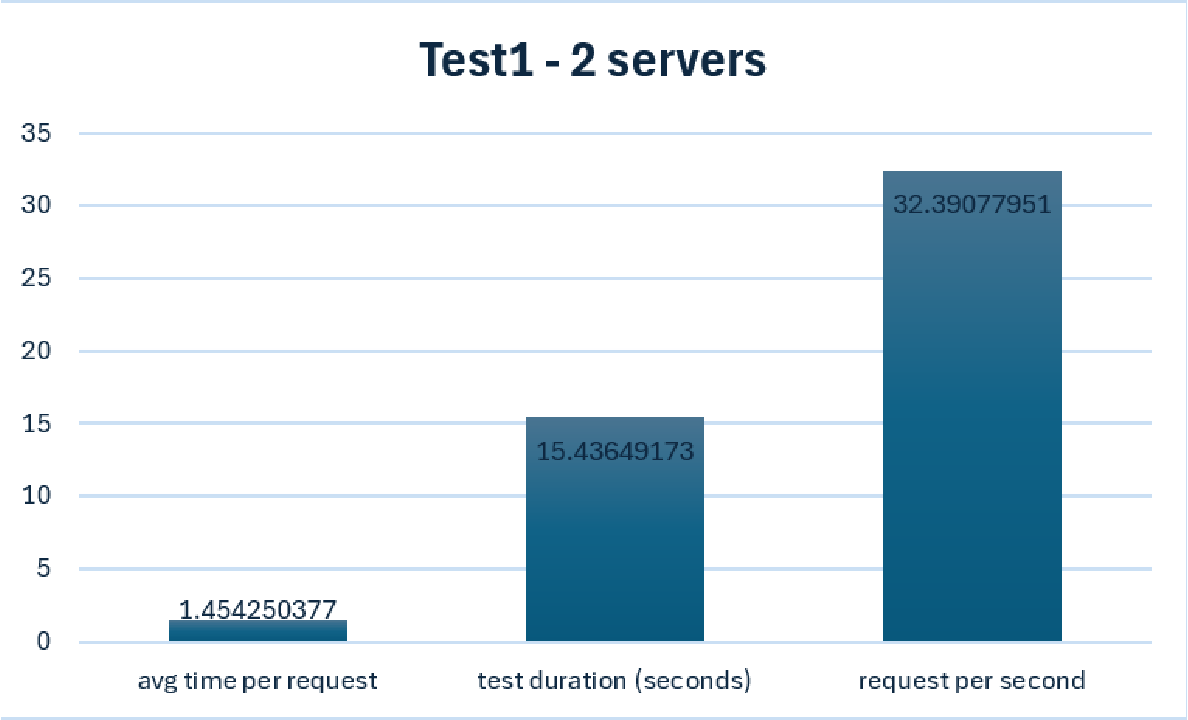
Evaluation

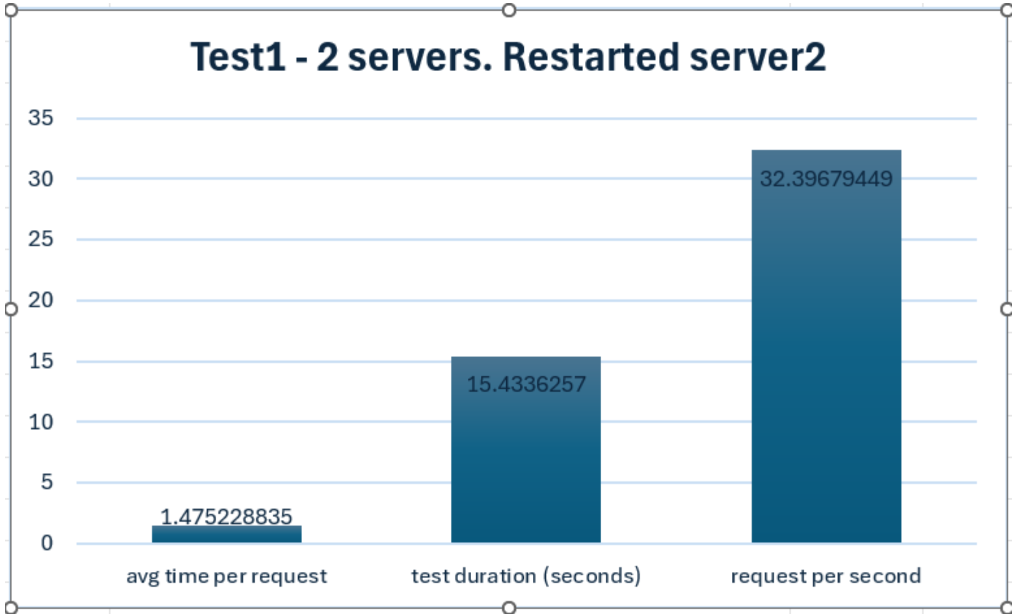
- Done with a python script. The script creates a number of threads, which simulate multiple clients accessing the resources.
- Can also be accessed with any client.
- A large enough prime number is selected for testing, so that the single container processing becomes the limiting factor. This will make the difference more clearly visible.
- Phase 1, first the tests are run with 2 servers on.
- Phase 2, server2 is stopped and health check is waited to prevent failed requests. Tests run with single server.
- Phase 3, server2 is restarted, health check is waited and tests run with 2 servers again.

Test 1

- Prime number 524287 selected.
  - A single request takes around 50ms with my machine.
- 50 threads each making 10 requests
- 500 successful requests

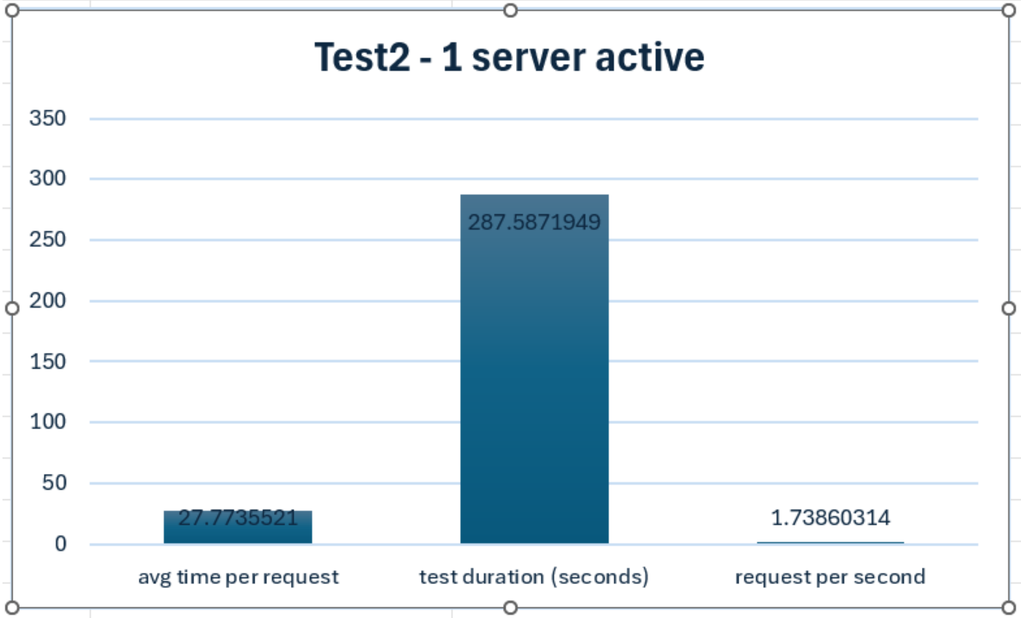


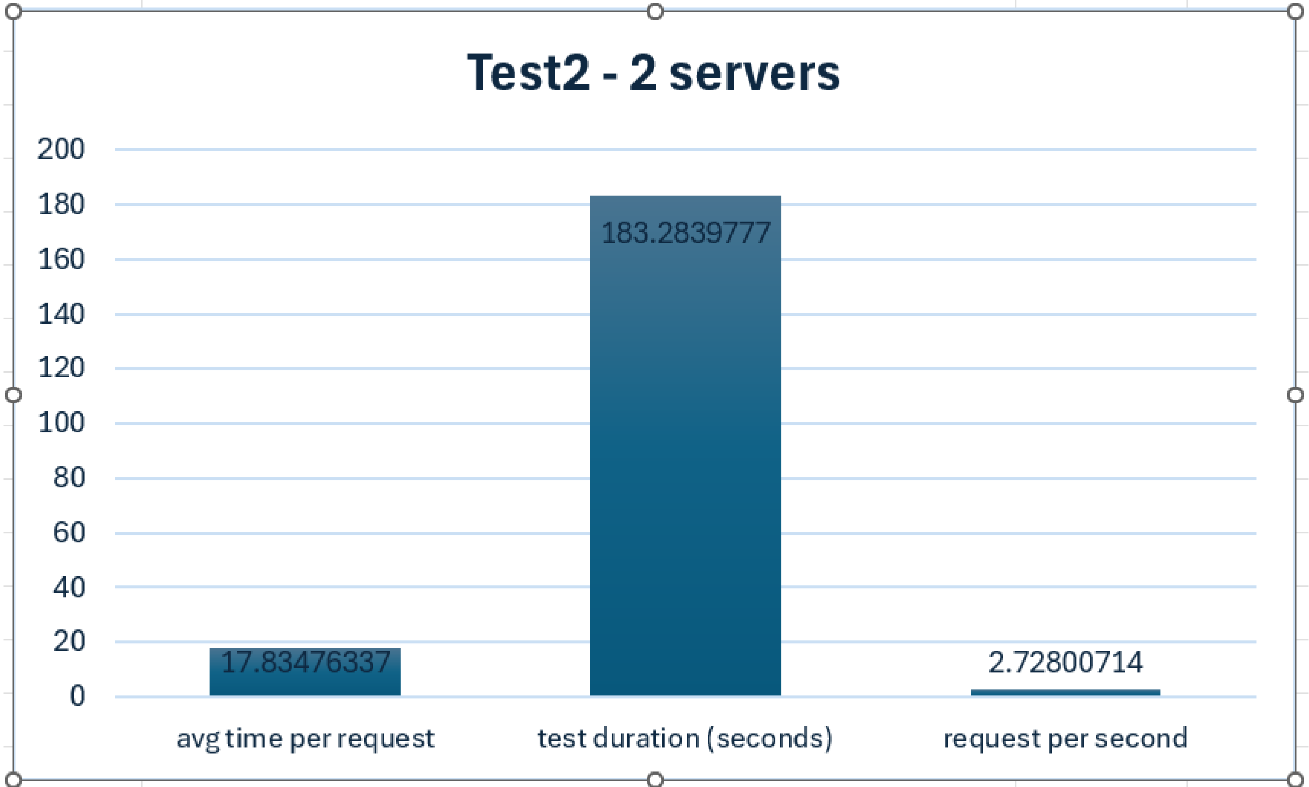


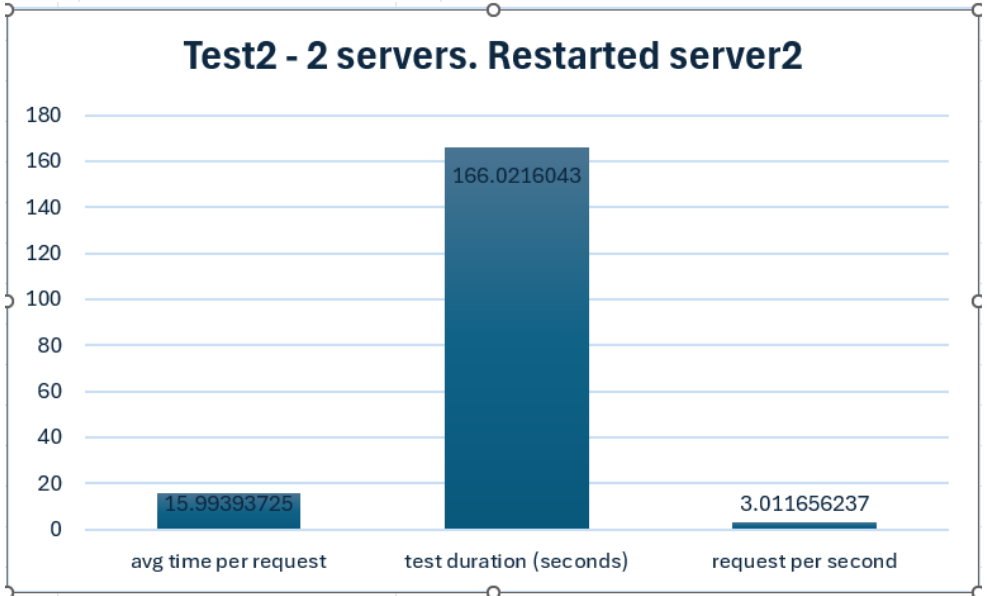


Test 2

- Prime number 6700417 selected.
  - A single request takes around 450ms with my machine
- 50 threads each making 10 requests
- 500 successful requests


Test2 - 1 server active


Test2 - 2 servers


Test2 - 2 servers. Restarted server2