

Testing Analysis

1. ****/rep endpoint `c`****

2. ****/heartbeat endpoint e.g for server at `http://localhost:5002/heartbeat` server_2****

```
nugi@DESKTOP-QVVGJA3:~/DistributedSysServer$ curl -X GET http://localhost:5002/heartbeat
nugi@DESKTOP-QVVGJA3:~/DistributedSysServer$ curl -X GET http://localhost:5000/rep
{
  "message": {
    "N": 3,
    "replicas": [
      "server_2",
      "server_1",
      "distributedsysserver-load_balancer-1"
    ]
  },
  "status": "successful"
}
```

3. ****/home endpoint e.g for server at `http://localhost:5001/home` server_1****

```
nugi@DESKTOP-QVVGJA3:~/DistributedSysServer$ ls
Makefile  code_test  docker-compose.yml  load_balancer_requirements  server_requirements
nugi@DESKTOP-QVVGJA3:~/DistributedSysServer$ curl -X GET http://localhost:5001/home
{"message": "Hello from Server: 1", "status": "successful"}
nugi@DESKTOP-QVVGJA3:~/DistributedSysServer$
```

4. ****/add endpoint****

Provide the n field and a list of hostnames e.g adding 4:

curl -X POST http://localhost:5000/add -H "Content-Type: application/json" -d '{

"n": 4,

"hostnames": ["server_4", "server_5", "server_6", "server_7"]

}'

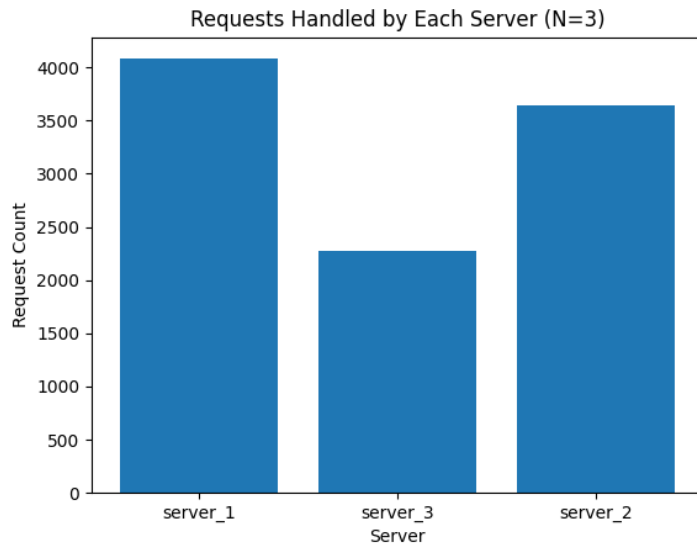
```
nugi@DESKTOP-QVVGJA3:~/DistributedSysServer$ curl -X POST http://localhost:5000/add -H "Content-Type: application/json" -d '{
  "n": 4,
  "hostnames": ["server_4", "server_5", "server_6", "server_7"]
}'
{
  "status": "successful"
}
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
NAMES					
e2d3ab48640c	myproject_server	"python server.py"	22 hours ago	Created	
server_8					
4a0c956f46e1	myproject_server	"python server.py"	22 hours ago	Created	
server_3					
fa86b3bd2ec	myproject_server	"python server.py"	22 hours ago	Created	
server_7					
e01eec4aee9	myproject_server	"python server.py"	22 hours ago	Created	
server_6					
79aed9ac6a8a	myproject_server	"python server.py"	22 hours ago	Created	
server_5					
2ad3bbf4d9e7	myproject_server	"python server.py"	22 hours ago	Created	
server_4					
2e1b05f7515d	myproject_server	"python server.py"	22 hours ago	Up 22 hours	0.0.0.0:5002->5000/tcp, :::5002->5000/tcp
server_2					
f5de625ceafa	myproject_server	"python server.py"	22 hours ago	Up 22 hours	0.0.0.0:5001->5000/tcp, :::5001->5000/tcp
server_1					
587d4facebe25	myproject_load_balancer	"python load_balance..."	22 hours ago	Up 22 hours	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
distributedsysserver-load_balancer-1					

Testing Load balancing

Load Distribution Among 3 Servers**

Observations

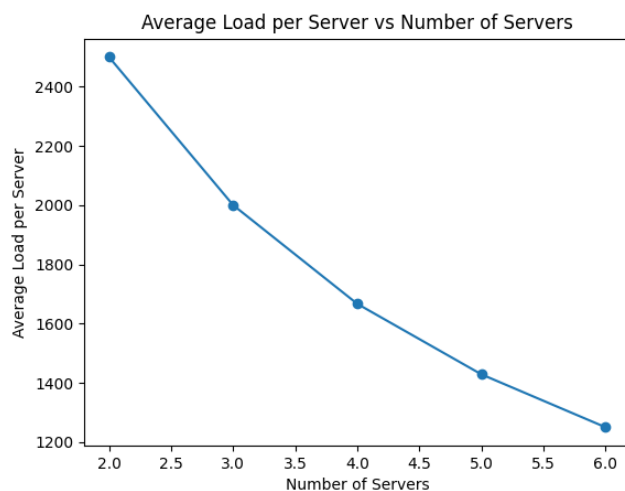


Analysis

- The load distribution is uneven, with `server_1` handling the most requests and `server_3` handling the least.
- Possible reasons for this discrepancy could include the network latency, or environmental factors.

Scalability with Incrementing Servers N from 2 to 6**

Observations



Analysis

- The average load per server decreases as the number of servers increases.
- The load balancer scales efficiently with more servers.

Load Balancer Recovery from Server Failure

Observations

```
eric@eric-VirtualBox:~/loadbalancing$ python3 testing_code/fail.py
Starting Failure Handling Test
Add Servers: {'message': {'N': 7, 'replicas': ['server_7', 'server_6', 'server_5', 'server_4', 'server_2', 'server_1', 'server_3']}, 'status': 'successful'}
Sending initial requests
Request: home, Request ID: 318016, Response: {'message': '{"message": "Hello from Server: server_4", "status": "successful"}\n', 'server': 'server_4'}
Request: home, Request ID: 452944, Response: {'message': '{"message": "Hello from Server: server_6", "status": "successful"}\n', 'server': 'server_6'}
Request: home, Request ID: 351207, Response: {'message': '{"message": "Hello from Server: server_6", "status": "successful"}\n', 'server': 'server_6'}
Request: home, Request ID: 533055, Response: {'message': '{"message": "Hello from Server: 1", "status": "successful"}\n', 'server': 'server_1'}
Request: home, Request ID: 706113, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 125788, Response: {'message': '{"message": "Hello from Server: server_6", "status": "successful"}\n', 'server': 'server_6'}
Request: home, Request ID: 423448, Response: {'message': '{"message": "Hello from Server: server_5", "status": "successful"}\n', 'server': 'server_5'}
Request: home, Request ID: 777579, Response: {'message': '{"message": "Hello from Server: server_7", "status": "successful"}\n', 'server': 'server_7'}
Request: home, Request ID: 902926, Response: {'message': '{"message": "Hello from Server: server_7", "status": "successful"}\n', 'server': 'server_7'}
Request: home, Request ID: 66447, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Removing some servers
Remove Servers: {'message': {'N': 5, 'replicas': ['server_7', 'server_6', 'server_4', 'server_2', 'server_1']}, 'status': 'successful'}
```

```
Remove Servers: {'message': {'N': 5, 'replicas': ['server_7', 'server_6', 'server_4', 'server_2', 'server_1']}, 'status': 'successful'}
Sending requests after removing servers
Request: home, Request ID: 574236, Response: {'message': '{"message": "Hello from Server: 1", "status": "successful"}\n', 'server': 'server_1'}
Request: home, Request ID: 442655, Response: {'message': '{"message": "Hello from Server: 1", "status": "successful"}\n', 'server': 'server_1'}
Request: home, Request ID: 193810, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 540122, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 973290, Response: {'message': '{"message": "Hello from Server: server_7", "status": "successful"}\n', 'server': 'server_7'}
Request: home, Request ID: 879465, Response: {'message': '{"message": "Hello from Server: server_4", "status": "successful"}\n', 'server': 'server_4'}
Request: home, Request ID: 225964, Response: {'message': '{"message": "Hello from Server: server_7", "status": "successful"}\n', 'server': 'server_7'}
Request: home, Request ID: 664145, Response: {'message': '{"message": "Hello from Server: server_7", "status": "successful"}\n', 'server': 'server_7'}
Request: home, Request ID: 382761, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 205191, Response: {'message': '{"message": "Hello from Server: server_7", "status": "successful"}\n', 'server': 'server_7'}
Starting Scaling Down Test
Add Servers: {'message': {'N': 11, 'replicas': ['server_13', 'server_12', 'server_11', 'server_10', 'server_9', 'server_8', 'server_7', 'server_6', 'server_4', 'server_2', 'server_1']}, 'status': 'successful'}
Sending initial requests
Request: home, Request ID: 686588, Response: {'message': '{"message": "Hello from Server: server_8", "status": "successful"}\n', 'server': 'server_8'}
```

```
: 'successful'}
Sending initial requests
Request: home, Request ID: 686588, Response: {'message': '{"message": "Hello from Server: server_8", "status": "successful"}\n', 'server': 'server_8'}
Request: home, Request ID: 930604, Response: {'message': '{"message": "Hello from Server: server_4", "status": "successful"}\n', 'server': 'server_4'}
Request: home, Request ID: 884282, Response: {'message': '{"message": "Hello from Server: server_10", "status": "successful"}\n', 'server': 'server_10'}
Request: home, Request ID: 842585, Response: {'message': '{"message": "Hello from Server: server_9", "status": "successful"}\n', 'server': 'server_9'}
Request: home, Request ID: 525900, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 847064, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 655926, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Request: home, Request ID: 277382, Response: {'message': '{"message": "Hello from Server: server_8", "status": "successful"}\n', 'server': 'server_8'}
Request: home, Request ID: 384655, Response: {'message': '{"message": "Hello from Server: server_10", "status": "successful"}\n', 'server': 'server_10'}
Request: home, Request ID: 879695, Response: {'message': '{"message": "Hello from Server: 2", "status": "successful"}\n', 'server': 'server_2'}
Scaling down by removing servers
Remove Servers: {'message': {'N': 7, 'replicas': ['server_13', 'server_12', 'server_11', 'server_10', 'server_9', 'server_4', 'server_2']}, 'status': 'successful'}
Sending requests after scaling down
Request: home, Request ID: 591363, Response: {'message': '{"message": "Hello from Server: server_10", "status": "successful"}\n', 'server': 'server_10'}
Request: home, Request ID: 879149, Response: {'message': '{"message": "Hello from Server: server_10", "status": "successful"}\n', 'server': 'server_10'}
Request: home, Request ID: 868839, Response: {'message': '{"message": "Hello from Server: server_12", "status": "successful"}\n', 'server': 'server_12'}
Request: home, Request ID: 387560, Response: {'message': '{"message": "Hello from Server: server_1", "status": "successful"}\n', 'server': 'server_1'}
```

Initial Requests with addition of 4 Servers

server_4: 1 request

server_6: 3 requests

server_1: 1 request

server_2 : 2 requests

server_5: 1 request

server_7: 2 requests

Post-Failure Requests with deletion of 2 Servers

server_1: 2 requests

server_2: 3 requests

server_7: 4 requests

server_4: 1 request

Observations:

- The load balancer quickly detected the removal of 2 servers and redistributed the load.
- The response times remained stable, indicating efficient handling of server failures.

Scaling Down Test

Initial Requests with addition of 6 Servers

server_8: 2 requests

server_4: 1 request

server_10: 2 requests

server_9: 1 request

server_2: 4 requests

Post-Scaling Requests with removing 2 Servers

server_10: 2 requests

server_12: 2 requests

server_11: 2 requests

server_9: 2 requests

server_2: 3 requests

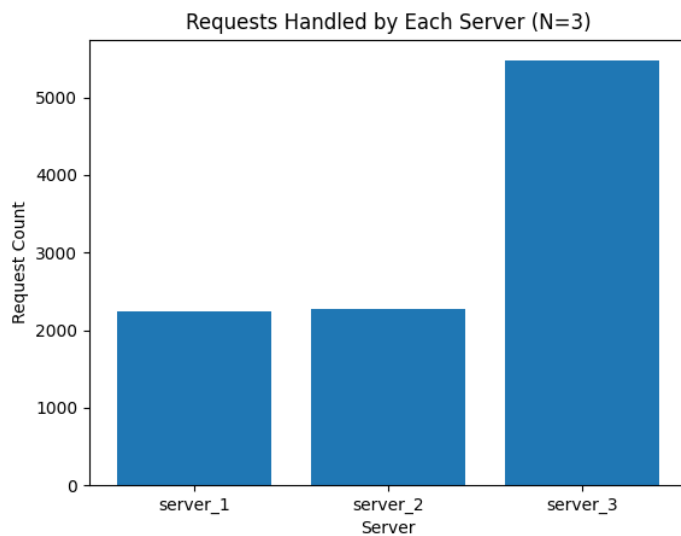
Observations:

- The load balancer effectively scaled down when deletion of 2 servers.
- The load distribution post-scaling was balanced, and the system maintained the performance.

Testing Load balancing for A-4 modifying the hash and virtual server functions using md5

Load Distribution Among 3 Servers

Observations

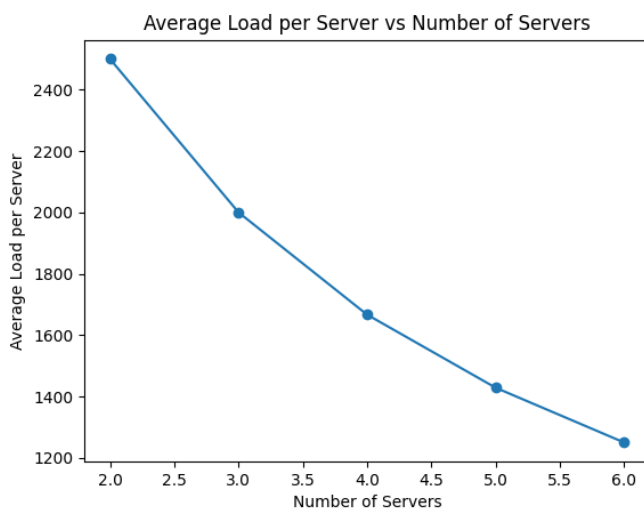


Analysis

- The MD5 hash function resulted in an imbalanced load distribution among the servers, with one server handling a significantly higher number of requests.
- The load distribution with the MD5 hash function is less balanced compared to the original SHA-256 based hash function.
- This imbalance indicates that the MD5 hash function might not be as effective in distributing requests evenly across servers.

A-2 Scalability with Incrementing Servers N from 2 to 6

Observations



Analysis

- The average load per server decreases as the number of servers increases, indicating good scalability.
- Despite the imbalances observed in A-1, the system scales well with the MD5 hash function, distributing the load across an increasing number of servers.