

Tribhuvan University
Institute of Engineering
Pulchowk Campus

INTERNET AND INTRANET

Lab 4

Adaptive Load Balancing

SUBMITTED BY:

Bishal Katuwal
075BCT028

SUBMITTED TO:

Department of Electronics and Computer Engineering
Pulchowk Campus

SUBMITTED ON:

7th February, 2023

Title

Adaptive Load Balancing

Background Theory

Load balancing is the process of distributing incoming traffic across multiple devices based on various factors, such as the number of connections, the processing load on each resource, or the response time of each resource. It is done to optimize resource utilization, minimize response time, and avoid overloading any single resource.

Adaptive load balancing is a method of load balancing used to to optimize resource utilization and maximize the processing of incoming requests. The goal of adaptive load balancing is to dynamically adjust to changes in traffic patterns and server utilization, allowing the system to respond quickly and efficiently to changing conditions.

Activity The following activity depicts simulation of adaptive load balancing in a system with 5 servers and 50 requests.

```
import random

# Defines a server with its unique ID
class Server:
    def __init__(self, id):
        self.id = id
        self.utilization = 0
    def __str__(self):
        return('Server'+str(self.id)+ ' = ' +str(self.utilization))
    def __repr__(self):
        return str(self)

# Load Balancing Algorithm
class LoadBalancer:
    def __init__(self, resources):
        self.resources = resources
    def allocate_request(self, request):
        resource = min(self.resources, key=lambda x: x.utilization)
        resource.utilization += request
        return resource.id
    def release_request(self, request, resource_id):
        resource = next(r for r in self.resources if r.id == resource_id)
        resource.utilization -= request

if __name__ == "__main__":
    # Generate 5 servers
    servers = [Server(i) for i in range(5)]
    AptLoadBal = LoadBalancer(servers)

    # Generate requests
    requests = [random.randint(1, 10) for i in range(50)]
```

```
# Allocate requests to resources
resource_ids = [AptLoadBal.allocate_request(r) for r in requests]

# Release requests from resources
for i, r in enumerate(requests):
    AptLoadBal.release_request(r, resource_ids[i])
print(servers)
```

Conclusion

In this report, we have discussed the basic steps for adaptive load balancing in network system. The process of adaptive load balancing involves finding the server with lowest utilization and allocating the request to that node.

Appendix

Roll No. 28

Topic = (Roll No % 8) + 1

= (28%8)+1

=5th topic