COMP 6231: Distributed Systems Design

Lab 6 Instructions

Load Balancer

June 12, 2022

1 General Information

Lab Date: Tuesday, June 14th, 2022.

Your lab instructor will provide you with instructions on how to do this lab activity.

2 Overview

The purpose of this lab is to obtain a hands-on experience on load balancing on server side when there is a traffic on network. In this lab you are expected to enhance provided Server, Client and LoadBalancer server code to complete a load balancing on server side. (Optional) Students are welcome to develop their own solution from scratch.

3 Load Balancer

A load balancer acts as the “traffic cop” sitting in front of your servers and routing client requests across all servers capable of fulfilling those requests in a manner that maximizes speed and capacity utilization and ensures that no one server is overworked, which could degrade performance.

In this manner, a load balancer performs the following functions:

1. Distributes client requests or network load efficiently across multiple servers
2. Ensures high availability and reliability by sending requests only to servers that are online
3. Provides the flexibility to add or subtract servers as demand dictates

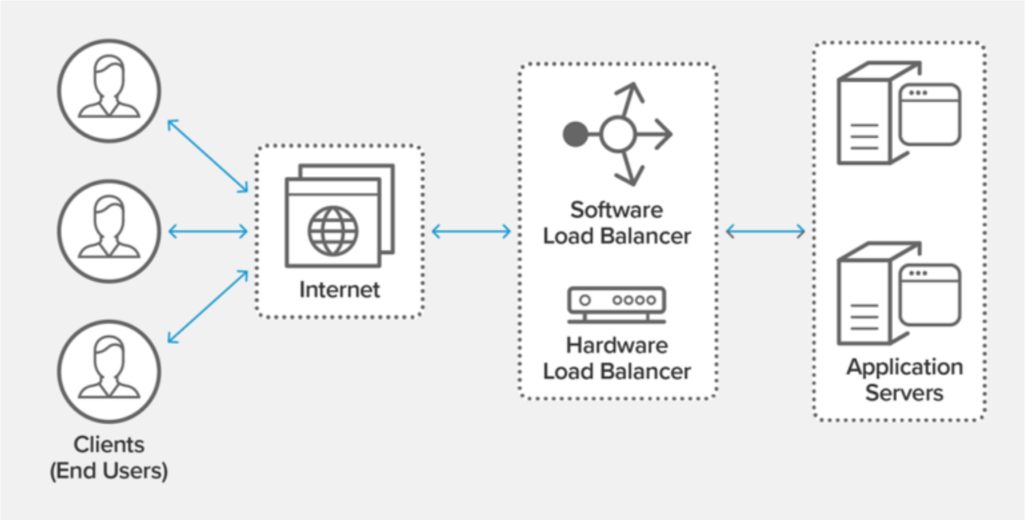

figure 1: load balancing diagram

4 Instructions

You may start from scratch or use provided lab templates.
Clone the template from the second tutorial on Github:
`https://github.com/COMP6231/T06`
you can see the program files for your better understanding if you are starting from scratch. if you are using base code then you need to complete RoundRobin.java, Task.java and LoadBalancer.java file to complete lab exercise.

4.1 RoundRogin

Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way. It is basically the preemptive version of First come First Serve CPU Scheduling algorithm.

your task in this lab is to complete the RoundRobin.java file by implementing RoundRobin Algorithm.

4.2 LoadBalancer

Generally, LoadBalancer will act as a central server and handles multiple server's load.So in this lab, we are going to implement that. We have 4 servers that are running on port 6001..6004. When client connect with LoadBalancer, it will assign a free server or a server that has less number of client connected with. We will create a new task(Task class Instance) for every client when they connect with LoadBalancer and will pass client socket and port number of the free server using RoundRobin algorithm as a parameters. For every new Task, you must start new Thread.

4.3 Task

You have to complete task.java file too. This file will require your more effort for establishing a connection between 2 different port. you have to define TCP connection with the server. one Connection is already established between Task and Echoclient. Now, you have to define TCP connection between Task and server file. for that you will need a port number of the server to define a socket and stream. So,you will find a port number as a class variable(i.e portServer).Once, you define a TCP connection between Task and Server. you have to make communication between EchoClient, Task and Server using stream. Overview of that is when client sends the mathematical equation, it will pass to server by task and then when server sends reply of that, it will again pass to client back by task.

4.4 Client

EchoClient.java file is provided for you with full implementation. In which we are generating mathematical equations and sending them to the server for processing. we do receive a response from a server for each equation.

4.5 Server

Server.java file is also implemented for you with full implementation. In which, server is receiving a equation to process from task.java file and after that server will use problem.java file to solve that equation. Once the result is found for that equation, server will send that result back.

4.6 Problem

Problem.java file is also implemented for you with full implementation. In which, we are implemented 2 methods. one is for generating mathematical equations and other is to solve those equations. this program is used by Client and Server.

5 Evaluation

By completing all the classes, you should be able to run multiple servers on different port and there is one load balancer server which will act as a central server and handles the other servers running on different port using RoundRobin algorithm. So, Firstly, Client connects with central server and then it will assign a new server using RoundRobin algorithm. Then as per our assignment, Client will send mathematical equations and server will respond to them. At the last, you should be able to run loadbalancer.bat file successfully. If you don't want to use loadbalancer.bat file, you can run by yourself too.

6 After the Lab

you can explore the sticky session and try to implement that in this code.

With sticky sessions, a load balancer assigns an identifying attribute to a user, typically by issuing a cookie or by tracking their IP details. Then, according to the tracking ID, a load balancer can start routing all of the requests of this user to a specific server for the duration of the session.

References

1. A Guide to Load Balancer:

   `https://www.nginx.com/resources/glossary/load-balancing`

2. A Guide to RoundRobin algorithm:

   `https://www.geeksforgeeks.org/program-round-robin-scheduling-set-1`

3. Establishment of TCP connection:

   `https://www.baeldung.com/a-guide-to-java-sockets`

4. Sticky session:

   `https://www.imperva.com/learn/availability/sticky-session-persistence-and-cookies`