Armen Melkumyan • 1st
Technical / Solutions Architect
1yr •

🔄 Load Balancer: Round Robin Algorithm 🔄

The Round Robin algorithm is one of the simplest methods of distributing client requests among a set of servers. In this method, the load balancer forwards a client request to each server in turn, distributing the requests evenly among the servers. It doesn't consider the current load or performance of the servers.

🛠️ Step-by-Step Implementation:

✅ Step 1: Initialize an array or list with the IP addresses (or identifiers) of the servers in the pool.

🔢 Step 2: Maintain a counter that tracks the index of the last server that was chosen.

➡️ Step 3: When a new request arrives, increment the counter and mod it by the number of servers to ensure it wraps around.

📤 Step 4: Forward the request to the server at the current index.

🔁 Step 5: Repeat Steps 3 and 4 for each incoming request.

#LoadBalancing #RoundRobinAlgorithm #NetworkEngineering

#ServerManagement #TechTips #Csharp

```
using System;
using System.Collections.Generic;

public class RoundRobinBalancer
{
    private List<string> servers = new List<string>();
    private int lastServerIndex = -1;

    public RoundRobinBalancer(List<string> serverList)
    {
        servers = serverList;
    }

    public string GetServer()
    {
        lastServerIndex = (lastServerIndex + 1) % servers.Count;
        return servers[lastServerIndex];
    }
}

public class RoundRobin
{
    public static void Main(string[] args)
    {
        var servers = new List<string> { "Server1", "Server2", "Server3" };
        var roundRobin = new RoundRobinBalancer(servers);

        Console.WriteLine(roundRobin.GetServer()); // Server1
        Console.WriteLine(roundRobin.GetServer()); // Server2
        Console.WriteLine(roundRobin.GetServer()); // Server3
        Console.WriteLine(roundRobin.GetServer()); // Server1 again
    }
}
```