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

🔄 Load Balancer: Least Connections 🔄

The Least Connections method directs new requests to the server with the fewest active connections. This strategy is more dynamic than Round Robin, as it considers the current state of each server, aiming to prevent overload.

🛠️ Step-by-Step Implementation:

✅ Step 1: Track the number of active connections for each server in a data structure.

🔢 Step 2: When a new request arrives, iterate through the list of servers to find the one with the least number of active connections.

➡️ Step 3: Forward the request to this server.

📤 Step 4: Update the server's active connection count accordingly.

🔁 Step 5: When a connection closes, decrement the active connection count for the respective server.

#LoadBalancing #LeastConnections #NetworkEngineering

#ServerManagement #TechTips #Csharp

```csharp
using System;
using System.Collections.Generic;
using System.Linq;

public class LeastConnectionsBalancer
{
    private Dictionary<string, int> serverLoad = new Dictionary<string, int>();

    public LeastConnectionsBalancer(List<string> servers)
    {
        foreach (var server in servers)
        {
            serverLoad[server] = 0;
        }
    }

    public string GetServer()
    {
        var server = serverLoad.OrderBy(s => s.Value).First().Key;
        serverLoad[server]++;
        return server;
    }

    public void ReleaseServer(string server)
    {
        serverLoad[server]--;
    }
}

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

        Console.WriteLine(leastConnections.GetServer()); // Server1
        Console.WriteLine(leastConnections.GetServer()); // Server2
        leastConnections.ReleaseServer("Server1");
        Console.WriteLine(leastConnections.GetServer()); // // Server1
    }
}
```