

ASSIGNMENT NO.3

TITLE: Load Balancing

OBJECTIVE: To implement hashing techniques

PROBLEM STATEMENT:

Load Balancing:

For example, imagine you have a set of servers that handle requests for a web application. The key to load balancing is using the hash value of a client's IP address or a request ID to determine which server should handle the request. The hash function is typically designed so that the data is evenly distributed across the servers, ensuring that no single server is overloaded. Write a program of a load balancing system

```
#include <iostream>
using namespace std;

int main() {
    int N, R, val, hash_val;

    cout << "Enter number of servers: ";
    cin >> N;

    cout << "Enter number of requests: ";
    cin >> R;

    int arr[N]; // To store request count per server
    for (int m = 0; m < N; m++) {
        arr[m] = 0;
    }

    for (int i = 0; i < R; i++) {
        cout << "Enter request " << i + 1 << ": ";
        cin >> val;

        hash_val = val % N;

        // Assign request directly to the hashed server
        arr[hash_val]++;
    }
}
```

```

        cout << "Request " << val << " assigned to server " << hash_val << endl;
    }

    // Print final state of each server
    cout << "\nFinal request count per server:\n";
    for (int j = 0; j < N; j++) {
        cout << "Server " << j << ": " << arr[j] << " requests\n";
    }

    return 0;
}

```



A terminal window titled "Terminal" with standard macOS window controls (back, forward, search, and window management icons). The terminal displays the output of a C++ program. It starts with prompts for the number of servers (5) and requests (4). Then, it shows four requests being entered and assigned to servers: request 23 to server 3, request 43 to server 3, request 12 to server 2, and request 56 to server 1. After a blank line, it shows the final request counts for each server: Server 0 has 0 requests, Server 1 has 1 request, Server 2 has 1 request, Server 3 has 2 requests, and Server 4 has 0 requests. The program then exits with code 0, and the user is prompted to press return to continue, which is followed by a single 's' character.

```

Enter number of servers: 5
Enter number of requests: 4
Enter request 1: 23
Request 23 assigned to server 3
Enter request 2: 43
Request 43 assigned to server 3
Enter request 3: 12
Request 12 assigned to server 2
Enter request 4: 56
Request 56 assigned to server 1

Final request count per server:
Server 0: 0 requests
Server 1: 1 requests
Server 2: 1 requests
Server 3: 2 requests
Server 4: 0 requests

-----
(program exited with code: 0)
Press return to continue
s

```

 main

 1 Branch

 0 Tags


Go to file

t


Add file


<> Code


About

 SiddiqBagwan Add files via upload

479ad24 · 2 weeks ago

 3 Commits

 README



SYBtech_DS_lab

DS_practi

 Readme

 Activit

 0 stars

 0 watc

 0 fork

Releases

No releases

[Create a new release](#)

Packages

No packages

[Publish your package](#)

Language