EE450-Socket_Project

Personal Information

Full Name: ZHIYU HU

Student ID:3288624500

File Path

The backend server need to get data during the boot-up phase and the **data file paths**(they should be put into the same dir path of the project) are:

serverA: "./data1.txt"

serverB: "./data2.txt"

What I have done in the assignment

I have already finished requires phases in description:

Phase 1—Bootup:

- Backend server A and B read the files data1.txt and data2.txt respectively, and construct a list of "graphs"(using adjacency matrix);
- Main server asks backend servers for which countries they are responsible for;
- Main server construct a map to book-keep the country information;

Phase 2--Query:

- Clients send their queries to the Main server;
- After Main server receives the queries, it decodes the queries and decide which backends to handle these queries;

Phase 3--Recommendation:

- Main server sends query to the corresponding backend server;
- Backend server performs some operations based on the number of common neighbors(using adjacency matrix) to do recommendations;
- Backend servers send the recommendations result back to Main server ;

Phase 4--Reply:

- Main server decodes the messages from Backend servers ;
- Main server prepares a reply message and sends it to the Client;
- Clients receive the recommendation from Main server and display the corresponding information

Code Files and their functions

- **client.cpp:**

  Code for Client to communicate with Main server by TCP.

  1. Ask for user to input the userID and country;
  2. Send the query to Main server;
  3. Get the recommendation result from Main server and print it out on the screen;
- **servermain.cpp:**

  Code for Main server which have following functions:

  1. Receives the query message(userID and country) from clients with TCP;
  2. Get country lists corresponds to backend serverA&serverB during boot-up phase and store them the local map;
  3. Check whether the country the client input is in the country map;
  4. If the country exists, send query message to corresponding backend server with UDP;
  5. Receives recommendation result from backend server and send back to client;
- **serverA.cpp**

  Code for backend server A connected with Main server by UDP.

  1. Read the data1.txt file and construct the graphs;
  2. Send country in data1.txt back to Main server;
  3. Receives the query message(userID and country) from Main server and searches if this userID is in the corresponding country map;
  4. If available, send the recommendation back to Main server.
- **serverB.cpp**

  Code for backend server B connected with Main server by UDP. It is almost the same with serverA.cpp except the UDP port number;

Format of message exchange

The message print on screen is the same as the requirement in project description;

And the message exchange during each phase:

Phase 1—Bootup:

- The Main server sends message **"bootup"** to both backend servers;
- The backend serverA and serverB get the message **"bootup"** and send the corresponding country list using format like "US | Japan | Canada | China";

Phase 2--Query:

- The Client sends the message **"UserID | Country"**(e.g. "12 | US") to Main server;

Phase 3--Recommendation:

- case1: If the country is not available in the country map, the Main server sends the message **"CountryNF"** to clients;
- case2: If the country is available in the country map, then the Main server sends the query message **"UserID | Country"** to corresponding backend server;

Phase 4--Reply:

For backend server:

- case1: If the userID is not found in the corresponding graph, then the backend server sends the message **“userIDNF”**;
- case2: If the userID is found while there are only user in the graph or the user is connected to all the other users, then the backend server sends the message **“None”**;
- case3: Otherwise, the backend server will do some operations and sends the result **userID**(e.g. “123”);

For Main server:

It will send the same recommendation message from backend server to clients (i.e. **“userIDNF”** or **“None”** or the **UserID**)

Idiosyncrasy in project

The max length of buffers are set to 1024. If the query message exceeds this size , the program will crash. (It is **not possible** in this project for the assumption: “The length of the name can vary from 1 letter to at most 20 letters.”)

Resued Code

1. The **implementation of TCP and UDP** (such as “ create socket” , “bind()”, “sendto()”, “recvfrom()”…) refers to the “Beej’s Guide to Network Programming” tutorial;
2. The use of fork() function to deal with multiple clients refers to a video in youtube(https://www.youtube.com/watch?v=BIJGSQEipEE);