

Distributed Mutual Exclusion – Implementation of Ricart-Agrawala Algorithm

Kareem Lawal

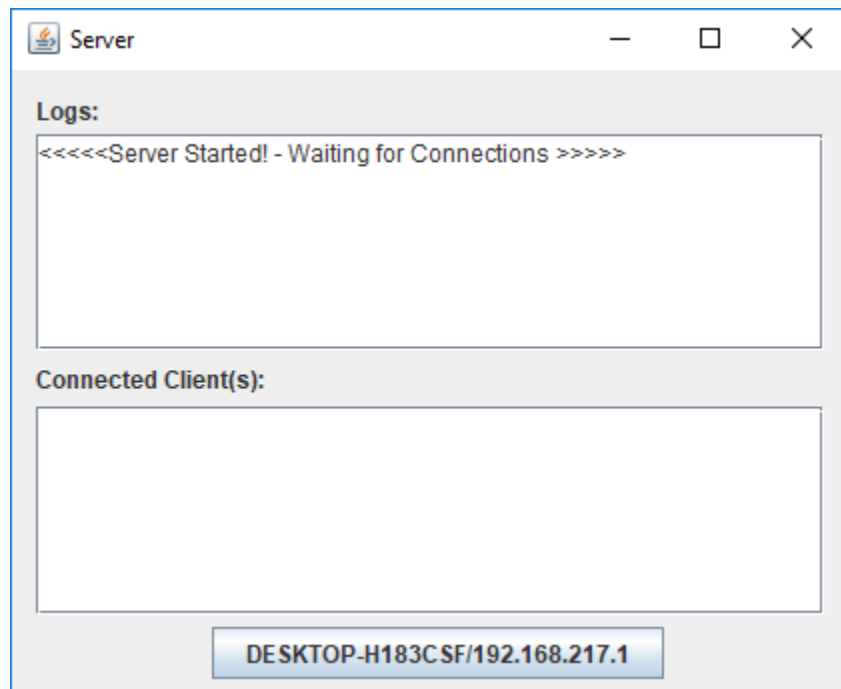
This program is an implementation of Distributed Mutual Exclusion using Ricart-Agrawala Algorithm. (Reference: https://en.wikipedia.org/wiki/Ricart-Agrawala_algorithm)

The program represents how timestamped processes enter a critical section after receiving all reply messages and upon exiting the critical section, the process broadcasts all deferred reply messages to other processes that are waiting to enter the critical section.

This program was designed using socket programming in Java. The critical section represents a spot where a client can request capitalization and sorting services from the server.

Instructions:

Run the server (CapSortServer_3.jar). This trigger opening of port 8888 for client's connections.



Server Interface

Run the client application and input the server IP address (Localhost if both server and client are running on the same host computer) and port 8888 (Server Port) then hit connect.

Client

CAPITALIZATION AND SORTING CLIENT

Server IP: Port: N:

String Input:

Client Interface

Each connected client is assigned a timestamp and the client with the maximum timestamp enters the critical section to request services from the server. Any client not in the critical section is denied the server services.

The program addresses node failure which is one of the major problem with the Ricart-Agrawala algorithm. Once a node in the critical section fails, it exits the critical section and the next client with the highest timestamp then enters the critical section.

In the sample run below, Client 3 has the highest timestamp as seen in the right-hand side of the client interface therefore allowing Client 3 to enter the critical section to request a string to be capitalized and sorted. Meanwhile, Client 1 and 2 capitalization and sorting request were rejected.

Client

CAPITALIZATION AND SORTING CLIENT

Server IP: Localhost Port: 8888 N:

String Input: Capitalization and Sorting

Send

Connected

Client 1 : 0.6108187669135523
Client 2 : 4.782109656389741
Client 3 : 9.401640825308826

Hello, you are client :3

Response from Server:
AND
CAPITALIZATION
SORTING

Client 1 : /127.0.0.1 : 58041
Client 2 : /127.0.0.1 : 58134
Client 3 : /127.0.0.1 : 58135

Client 3 request processed by the Server

The Figures below shows requests from Clients 1 and 2 being rejected.

Client

CAPITALIZATION AND SORTING CLIENT

Server IP:

Localhost

Port:

8888

N:

String Input:

Capitalization and Sorting

Send

Hello, you are client :1

Response from Server:
You are not in the Critical Section

Connected

Client 1 : 0.6108187669135523
Client 2 : 4.782109656389741
Client 3 : 9.401640825308826

Client 1 : /127.0.0.1 : 58041
Client 2 : /127.0.0.1 : 58134
Client 3 : /127.0.0.1 : 58135

Client

CAPITALIZATION AND SORTING CLIENT

Server IP:

Localhost

Port:

8888

N:

String Input:

Capitalization and Sorting

Send

Hello, you are client :2

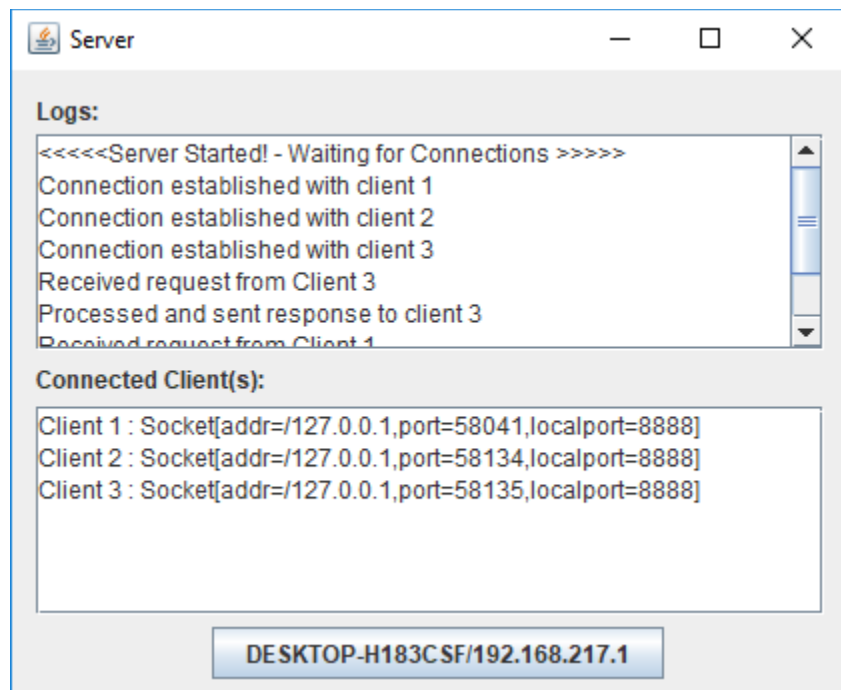
Response from Server:
You are not in the Critical Section

Connected

Client 1 : 0.6108187669135523
Client 2 : 4.782109656389741
Client 3 : 9.401640825308826

Client 1 : /127.0.0.1 : 58041
Client 2 : /127.0.0.1 : 58134
Client 3 : /127.0.0.1 : 58135

Client 1 and 2 being rejected services



Server Interface after three clients connected