CSCI-8780 Advanced Distributed Systems - Project 1: Distributed String Array with Concurrency Control (JAVA RMI)

List of Team Members

- Ratish Jha
- vatsal Thakkar

Prerequisites

Before you begin, make sure you have the following installed:

- Java Development Kit - 17 (JDK -17)

Getting Started

Follow these steps to set up and run the project:

Step 1: Open Three Terminals

Open three separate terminal windows or command prompts.

Step 2 & 3: Start Server and RMI registry using shell Script

you can start the server and RMI registry using the provided `server.sh` script. It will automatically start the RMI registry and Server.

```
chmod +x server.sh
./server.sh [-d]
-d for download local copy of java
```

Note: If you started the RMI registry and Server using script then you can skip the following two steps (`Step 2 (Manual) and Step 3 (manual)`) and directly go to `Step 4`.

Step 2: Start the RMI Registry (Manual)

In the first terminal, navigate to the 'server' directory:

```
cd server
```

Then, start the RMI Registry on port 9100 (You can use any port but make sure to change the port in `server-config.txt` and `client-config.txt` files):

Step 3: Start the Server (Manual)

Start Manually

```
rmiregistry 9100                              2/4
```

In the second terminal, navigate to the 'server' directory:

```
cd server
```

Compile the server code:

```
javac *.java
```

Start the server:

```
java Server server-config.txt
```

Step 4: Start the Client

In the third terminal, navigate to the 'client' directory

```
cd client
```

Compile the client code:

```
javac *.java
```

Start the client and interact with server:

```
java Client client-config.txt
```

Stop Server

```
chmod +x stop.sh
./stop.sh
```

Configuration File (server-config.txt & client-config.txt)

The `client-config.txt` & `server-config.txt` file is used to configure the client and server applications for interacting with a distributed string array. This file contains important information such as the server's host and port details, the array's initial capacity, and other configuration parameters.

Format (server-config.txt)

The `server-config.txt` file follows a specific format with each line serving a distinct purpose:

1. **Bind Name (rsa):** The first line specifies the bind name (`rsa`) under which the Remote String Array will be registered on the RMI registry. This name is crucial for the client to look up and communicate with the server's Remote String Array.

2. **Remote String Array Capacity:** The second line indicates the initial capacity of the Remote String Array. This value represents the maximum number of elements that can be stored in the array.

3. **Array Initialization:** The third line allows you to initialize the array with a list of comma-separated strings. These strings will populate the array's elements. If there are fewer strings than the specified capacity, the remaining elements will be initialized as `""(Empty String)`. This line is optional, and you can leave it empty if you don't want to initialize the array during startup.

4. **Host:** The fourth line specifies the IP address or hostname where the RMI registry is located. It's essential for the client to know the server's location to establish a connection.

5. **Port:** The fifth line designates the port number on which the RMI registry is listening. This port number is crucial for client-server communication.

Format (client-config.txt)

The `client-config.txt` file follows a specific format with each line serving a distinct purpose:

1. **Bind Name (rsa):** The first line specifies the bind name (`rsa`) under which the Remote String Array will be registered on the RMI registry. This name is crucial for the client to look up and communicate with the server's Remote String Array.

2. **Host:** The second line specifies the IP address or hostname where the RMI registry is located. It's essential for the client to know the server's location to establish a connection.

3. **Port:** The third line designates the port number on which the RMI registry is listening. This port number is crucial for client-server communication.

Example Configuration

Here's an example `server-config.txt` file:

```
rsa
5
Apple, Pineapple, Mango
```

```
127.0.0.1
9100
```

Here's an example `client-config.txt` file:

```
rsa
127.0.0.1
9100
```

Usage

Once the server and client are running, you can interact with the Distributed String Array with Concurrency Control. Follow the prompts in the client application to perform various operations on the distributed string array.