CS 5300 Project 1b
yl2458/ dj327/ wf85

I Overall Structure

Browser initiates the request and sends to the servlet, servlet parse the cookie and find the session information.
If session information is not found, a new session is created. Session is sent to RPC instracture and write to RPC server, waits for wq successful responses.
If session is found, then RPC client read from RPC server, it will update the session and write back.
Hashmap operation happens only after the RPC server received the information.

Cookies formats:
localServerID_rebootNum_sessionNumber_version_location(ServerIDs)

RPC messages:
UUID_OperationCode(R=0/=W1)_SessionID_SessionVersion_SessionMessage_ExpirationTime

II Functionality

1)Package RPC

RPC Client & RPC Server:
RPC Client contains ‘read’ and ‘write’ functions. In the ‘write’ function, ‘serverList’ records the server objects which are written message to. ‘repliedBricks’ is a subset of ‘serverList’. It represents the server objects which give responses to the RPC Client after writing operation. In the ‘read’ function, ‘queryMessage’ represents the format of the response RPC Server gives to RPC Client . ‘locations’ represents the target servers which response the read operation.
RPC Server keeps running in the whole process. ‘computeResponseFromRequest’ here is used to generate the response messages to RPC Client. The format of response is determined based on the value of ‘operationCode’.

RpcParameter:
RpcParameter stores parameters which are useful to RPC frame, F, R ,WQ, W; definatioon for operation code; two method for converting the format of information so that the information can fit the datagram pattern.

2)Package com.sessionmanagement

DataBrickManager:
Provides functions to retrieve ServerID.

ServerID:

Stores the ip address and port number of server.

Session:
Contains key parameters of session and constructors used to initialize a new session. Functions used to get and set the values of relevant parameters are also set here.

SessionManager:
This class contains functions to extract session information from cookie. It also responsible for generating a new session and storing session in a thread-safe HashMap. Plus, it cleans the timeout session from from the HashMap by 'CleanTask' class.

3)Script
Installation script:
The installation script has following functionality

1. Specify the number of instance to run
2. Specify the s3 bucket in which the war file is stored at
3. Configure the access key and security key
4. Download the war file and the reboot script to ec2 instance

Since the aws use old version of java, the script also delete the java7 and reinstall the java8
We get the local ip and ami index by the RESTFul function call.
The script will wait till three instance goes online.
Once these instance are online, we use "select * from serversData" to get all instance meta data in json format. The ami index is recorded and the reboot number is initialized to 0.

Runinstance script:

1.The script will configure the access key and security key.
2.clean the existing data base and creat the new database
3. Run three ec2 instance with specified parameter

Reboot script:
Reboot script will take the input number from the /home/ec2-user/reboot.txt, increase one and overwrite to the existing files.

4)WebContent File
Mock.html:
The first page when user open the browser, which has 3 button, 1 textfield and 1 label to represent the message.

Main.jsp:
This jsp file connects browser and SessionServlet, defines methods for each functions and create places for information such like cookie message, expiration time and so on. It transfers different requests to session servlet and put properly message from servlet to the browser.

Logout.html:
This file shown when user click “logout” button and displays logout message.

Error.html:
This file shown when program encounters exception and display error message.

III How to run:

0. Copy the ssm.war and reboot.sh file to the s3
1. Use bash command “`source runinstance.sh`” to start the file.
2. Register the public dns to bigdata.systems website
3. Open the browser and use command
   `server0.dj327.bigdata.systems:8080/ssm/SessionServlet`

4. To reboot the system
   -> Use the aws reboot functionality to reboot the system.
   -> Go to the home directory and execute the following command
   “`cd ~`”
   “`Sudo su`”
   “`Source reboot.sh`”