ASSIGNMENT 3

Submitted By-:

Yugaank Arun Sharma

Canvas Id – ysharma

Date of Submission: 11/04/2017

In the previous assignment, you developed a simple peer-to-peer distributed hash table for sharing bindings of keys to values. The protocol for retrieving bindings from the database is pull-based: a client (node) searches the network for the bindings for a key. In this assignment, you will extend this with a push-based protocol: Clients will be notified when new bindings are added for a key. The implementation of this push protocol will be based on the server-side events (SSE) API in the Jersey framework.

1. To simplify how the assignment works, when an application is started, it creates a state server object that registers it internally. When a Web service request arrives, the handler for that request looks up the state server in the main program state. Concurrent requests synchronize their accesses to the node's state using locks in the state server.
2. Then the server is started on that machine or instance in this case. It displays a cli. In this assignment, we cover listening on a item/key for any modifications (particularly addition in this assignment).
3. You can toggle listen on or off using listenOn and listenOff command. By listening on a key, if another node in the network adds a value to that key, the nodes listening on that key are notified. If you don't want to be notified, you can execute listenOff and any modifications made to that key will not be notified to that node which is not listening.
4. If you want to know, on which items or keys are you listening on, you can execute listeners command to know the keys on which you will be notified when any modification is done.
5. Operations like bindingAdditions, listenOn, listenOff, close event output stream, etc. were completed in the assignment. Similarly, for State, Node resources, node services and web client.
6. Binding additions are handled. Suppose an item is already at node 1, but when a new node joins the network, and if the item which was in node 1 should be transferred to new node, it is transferred.

You can see the testing that demonstrates all the operations between 3 instances joined together.

In the zip archive, there is README.PDF, dht-streaming.zip archive (assignment), video demonstrating the testing and dht.jar file (server address as localhost)