

CSE 486/586 Distributed Systems

Programming Assignment 4

Replicated Key-Value Storage

Components Used:

Replicated Key-Value Storage, uses following components:

1. 3 Buttons, <Put1, Put2, Put3>, these buttons are used to insert values, with their corresponding keys, in a <key,value> pair fashion. Keys are numbered from 0-19, and values vary depending on the buttons used.
2. LDump button used to display the <key, value> pairs stored in local storage.
3. Get is used to display the keys 0-19 along with their corresponding values.

In order to implement the replicated key-value storage, three classes are implemented namely, the Main Activity class, the Provider class, and the Click listener class.

Main Activity class:

Here the displaying pattern after the retrieval of key-value pairs, on click of LDump and Get buttons, is written. This class is instantiated first and the onCreate(), function has all the implementation of which class needs to be called upon, when a button is pressed or clicked.

ClickListener class:

This class is called by the Main Activity class, on button clicks (Put1, Put2, Put3), to insert values into ContentValues.

Provider class:

The Provider is basically used to store the key-value pair. These <Key, Value> pairs are inserted into the Content Provider, using the insert() function and are retrieved upon subsequent request using the Content Provider's query() interface. The Client-Server implementation is also performed in the Provider class.

onCreate(): Here, the server task and client task are instantiated.

Insert():

The insert() function initially checks for the current active AVD port, and each port should have a node id derived from its emulator port. This node id is obtained by applying the hash function to the emulator port. The returnVal() function, implemented makes a call to the hashing function, and the hash value obtained is compared with other ports. This function returns a boolean value, true, if the position or the AVD on which the key value pair to be inserted is obtained. Then the key-value pair is inserted into the file storage of the AVD obtained. Now, this key-value pair should be replicated, since the main

goal of implementation of this project is to perform replication of the data stored. The replication of key-value, is done on the other two AVDs. Also a check needs to be performed when the values are inserted, as the AVDs on which the replication is to be performed might crash, and be unresponsive, so regular check needs to be done, when replication of data is performed.

In order to achieve quorum replication, we need to ensure that the AVD or the coordinator which is sending the data to the replication ports receives an acknowledgement or minimum number of votes. If it does not receive an acknowledgement from the replicating ports, it should assume that the port is failed and should send the data to be replicated to the other active port.

To ensure that minimum number of votes are achieved, we use timeout, where the port sending data for replication must receive an acknowledgement within a specified period of time. The system current time is noted when the port sends data for replication to the replication ports, and continuous monitoring for timeout is done. If the acknowledgement is not received, it declares the port as failed, and now all the data is sent to the currently active port. Continuous monitoring needs to be performed so that when the dead port becomes active, it receives all the data, it missed due to failure.

Now since the port is dead, when it becomes active again, the data, it lost due to failure needs to be reassigned to that port. Thus failure recovery needs to be performed. When the failed node becomes active, it makes a call for recovery and thus the successor is called upon, which is responsible for assigning the values to the node, which became active.

Query():

The Query() interface of the content provider is used to obtain the key-value pairs upon click of LDump and Get buttons.

The LDump button upon click retrieves local values stored in all the AVDs and displays the key-value pairs. Here all the AVDs need to forward the key-value pairs stored local to them to the AVD, which makes a call to LDump.

The Get Button is used to get all the 20 key-value pairs (0-19), and display them.

The Consistency model followed is linearizable as all the values inserted into the system are visible all over the system instantaneously. Here, the insertion values is done instantly, and is also seperated by pauses of 1 sec. Thus linearizable model is followed.