Theory :

**Phase1:** The co-ordinator asks from each participator whether they have successfully completed their responsibilities for that transaction and are ready to commit. Each participator responds ‘yes |OK’ or ‘no|abort’.

Every participator writes its data records in a log. If it is unsuccessful to do, then it responds with a failure message; if it is successful, then it sends an OK message.

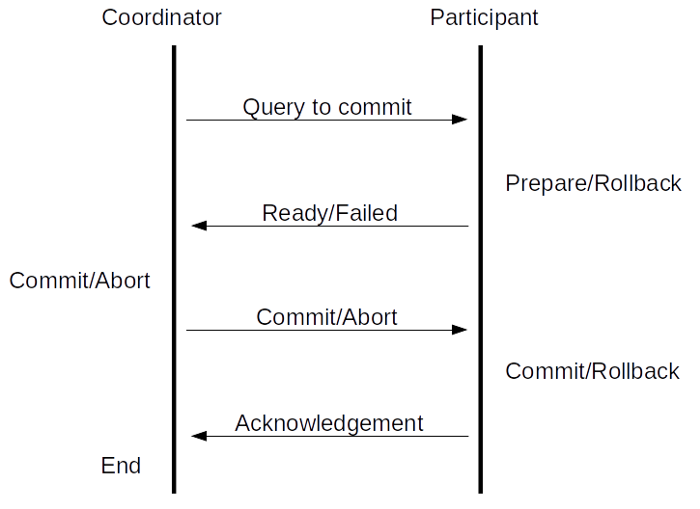
**Phase2:** This phase will start when Co-ordinator receive successful response in phase 1 from all participator.

Now Co-ordinator sends a commit request to all the participators.

Now participators write the commit as part of its log record.

Participants send a message that its commit has been successfully implemented.

If a server fails, the coordinator sends instructions to all servers to roll back the transaction.



Program Code :

**Serverside**

import java.io.\*;

import java.net.\*;

import java.lang.Thread;

import java.text.\*;

import java.util.\*;

public class serverside {

public static ArrayList<Thread> clients;

public static void main(String args[])throws Exception{

ServerSocket ss=new ServerSocket(5050);

int clientcount=1;

clients = new ArrayList<Thread>();

while(clientcount<=2){

Socket s=ss.accept();

System.out.println(clientcount+"connected");

DataInputStream din=new DataInputStream(s.getInputStream());

DataOutputStream dout=new DataOutputStream(s.getOutputStream());

ClientHandler ch = new ClientHandler(s,din,dout);

Thread t = new Thread(ch);

clients.add(t);

clientcount++;

}

for(int i=0;i<clients.size();i++) {

clients.get(i).start();

}

}

public static void abortall() {

int i=0;

System.out.println("Recieved a not ready signal \nAborting all processes");

}

}

class ClientHandler extends Thread {

static int i=0;

Socket s=null;

DataInputStream din;

DataOutputStream dout;

public ClientHandler(Socket s,DataInputStream din,DataOutputStream dout){

this.s = s;

this.din = din;

this.dout = dout;

}

public void run(){

try {

dout.writeUTF("<PREPARE T>");

String message = din.readUTF();

System.out.println(message);

if (message.equals("<NOT>")) {

//System.out.println("Alaikya");

new serverside().abortall();

}

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**Clientside**

import java.net.\*;

import java.io.\*;

import java.util.\*;

public class clientside {

public static void main(String args[])throws Exception{

Socket s=new Socket("localhost",5050);

DataInputStream din=new DataInputStream(s.getInputStream());

DataOutputStream dout=new DataOutputStream(s.getOutputStream());

String message = "";

message = din.readUTF();

System.out.println("Server sent "+message+"\n Press 1 to ready\n Press 0 to Not ready");

int flag = new Scanner(System.in).nextInt();

if (flag == 1) {

dout.writeUTF("<Ready T>");

}

else {

dout.writeUTF("<NOT>");

}

dout.close();

s.close();

}

}

Screenshots:

