677 lab2

Part 1:

Design

Multi-tier architecture

The server is constructed in multi-tier, which has a database layer and a front end layer consisting of two processes. The architecture is shown in the following figure.

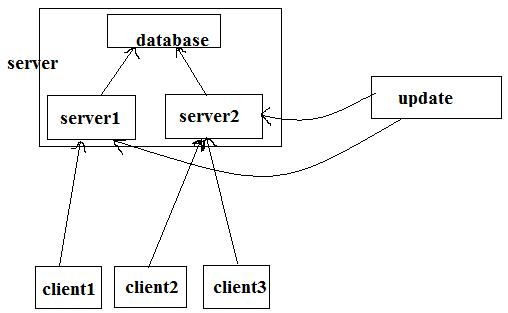


Figure 1 system architecture

1. Leader Election Algorithm: we use bully algorithm to implement leader election.

Figure 2 shows the process of the bully leader election algorithm. The database process has id=0, and there are two servers have id = 1 and id = 2. The server with id=1 initiate the leader election. It knows the id of database is lower than itself, so it only send message to server2 which has higher id. But since server2 is alive so it has the highest id, so it send messages to server1 and database claiming that it is the leader. This process happens when these server processes are booted. If the server2 initiate the election, the action in the left picture is not necessary since server2 don’t need to send message to processes with lower ids.

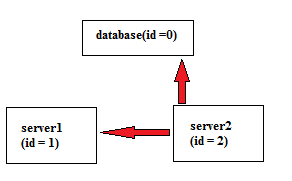
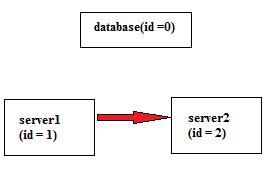


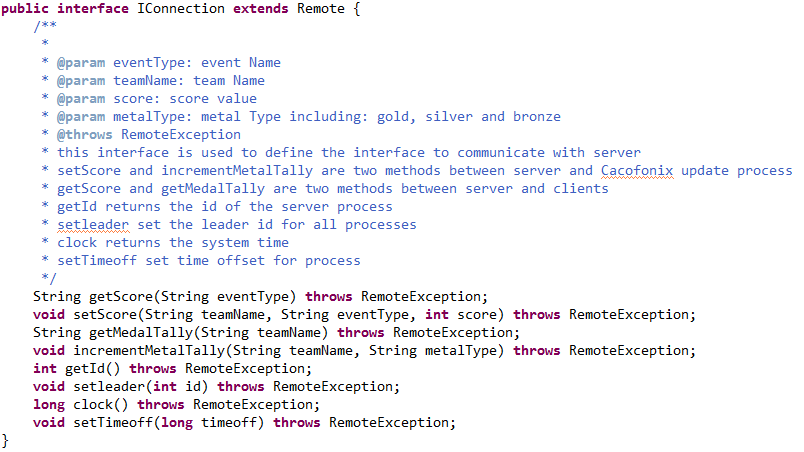
Figure 2 Bully algorithm for server processes

1. Clock Synchronization: we apply Berkeley Algorithm to adjust logical clock in our distributed system

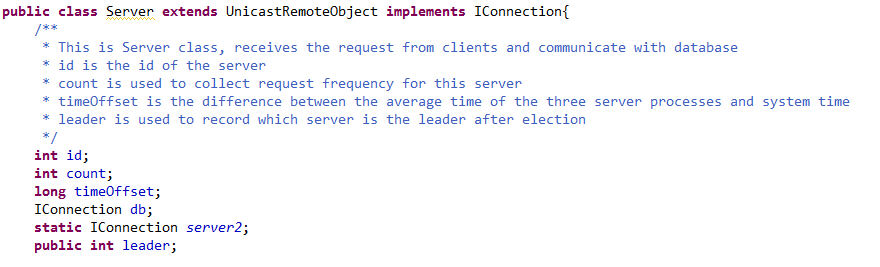
After we have chosen server2 as the leader of the three processes in the server, the leader should poll system time from other server processes and calculate the average time of the three ones. Then the leader should tell the slaves how much they should adjust (i.e. time offset)

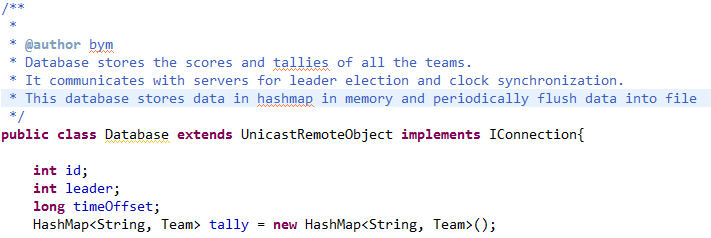
Class Implementation

We still use RMI to achieve the communications between client and server. In the server components, the front end servers also communicate with database in a nested client-server architecture. The interface for the communication is



Database and server implement this interface. Since they are in different layers, the implementations are different.





There is a thread in the server process which is in charge of implementing Berkeley Algorithm.

