15640 project4 documentation

Author: Qiyang He (qiyangh)

1. Server design

a. Server will hold a global list containing all the running tasks.
b. When there comes a new request, a new collage class will be build and added to the list.
c. When there come new messages, server will first detect which task it belongs to and then send to corresponding collage unit handling this message.
d. Each collage unit implements runnable and will run a thread after it is constructed. Such design enables handling concurrent incoming requests from multiple users.
e. When last ack is arrived, server will remove this task from global list, indicating it's a finished task.

2. User node design

a. User node will keep a global task list recording all the running tasks.
b. It will also lock resources if vote yes for a commit request.
c. When deciding vote for commit, if certain required resources has been deleted or locked, the user node will vote for no.
d. When receiving final decision from server, it will delete file or release lock on resources. Deleted file will also be recorded in case it will be required by future commit request.

3. Protocol between server and user node

a. I use a class containing necessary op code, task name, message target as well as other information to send message back and forth between server and user node.
b. Each time when sending the message, we need to serialize this message class to a byte array. Each time when receiving message, we need to deserialize this message to a normal message class.

4. Lost message handling and timeout thresholds

a. When server sending out commit propose to users, it will wait for 3 seconds. If any of the message timeout, which means there is no corresponding response from user, the commit will be abort.
b. When server sends out final commit decisions, it will wait for all the ack arrived, then remove task from global task list. Typically, server will send out a decision again each 3 second for users that are not responding.

5. Failure recovery and log design

a. Log design
    i. Generally, log record will have format like that:
        1. <collage name>,<op code>_<other op related params>
    ii. there will be 2 check points for user (ck2, ck4) and 3 check points for server (ck1, ck3, ck5). Below are example logs:
        1. 1.jpg,1_a:1.jpg_b:3.jpg_a:2.jpg
        2. 1.jpg,2_T_1.jpg_2.jpg
        3. 1.jpg,3_T_a_b
        4. 1.jpg,4_T_1.jpg_2.jpg
        5. 1.jpg,5_finish
b. server recovery
    i. if for a task there exists only 1 log (only ck1), server will send out abort commit decision to all the related users.
    ii. If for a task there exists 2 log, server need to distribute decision again to all the related users, and expected to receive ack from them.
    iii. If for a task there exists 3 log, server just ignore it (indicating a finished task).
c. User node recovery
    i. If for a task there exists 1 log (only ck2), user will lock related resources if necessary.
    ii. If for a task there exists 2 log, user need to update deleted resources list and also check local file storage to ensure the consistency of data.