CS 403 Assignment 3 Report
Gorkem Yar 27970

We expected to implement a synchronous Paxos mechanism in Python for this assignment. There are two files in my submission, paxos.py and sample_run.txt. The sample_run.txt file is an example of the Paxos algorithm that I implemented. In the following parts of this report, I will explain the algorithm and the functions I implemented in the paxos.py .

I implemented two broadcast mechanisms. The first one broadcasts the messages to all nodes except the sender. The other one broadcasts the messages to all the nodes with a failure probability. I also have two more networking functions named send and sendFailure. The former directly sends the message to the receiver socket, while the latter sends it with a failure probability.

The Paxos algorithm I implemented takes six parameters: ID, probability of failure, number of nodes, initial value for the node, number of rounds, and a barrier for synchronization. It first creates sockets using Zero Message Queue sockets. There are N (number of nodes) different ports, and each node has N+1 sockets for communication. One of the sockets of each node is a pull socket, which receives the messages from the unique port allocated for that node. The remaining sockets for a node are push sockets that are used for sending messages. After the initialization step, the Paxos algorithm runs in a for loop in the range of a given number of rounds. In each round, a leader is selected in a round-robin fashion. After the selection, there are two different executions for leader and non-leader nodes. Each execution consists of two different phases. At the end of the round, barriers force the synchronization in the algorithm.

The leader node consists of the START and PROPOSAL phases. In the START, the leader node uses the broadcastFailure method to notify all nodes about the new round and request their maxVoted information. If more than N/2 nodes answer this broadcast message with JOIN, then the PROPOSAL phase immediately starts; otherwise, the ROUNDCHANGE message is broadcasted to all nodes. In the proposal phase, the leader node finds the maxVotedValue (last voted value) and proposes it to the other nodes with the broadcastFailure method. If more than N/2 nodes answer this question with the VOTE message, then the leader node comes to a decision.

The non-leader node consists of two phases as well: JOIN and VOTE. In the JOIN phase, the non-leader nodes have to answer the coming START message with JOIN messages. The information of the maxVotedRound of a particular node is concatenated to the message. The non-leader nodes send this message to the leader node using the sendFailure method. In the second phase, if the leader node answers with PROPOSAL, non-leader nodes have to answer with the VOTE message by using the sendFailure method as well. Additionally, If the leader node sends CRASH, they will answer with CRASH. Finally, if the leader node sends ROUNDCHANGE, they will wait for the end of phase 2 in all of the nodes and immediately jump to the next round.