ECE 428 MP2 Design Documentation

Josiah McClurg and Yihua Lou

May 1, 2013

1 Basic Design

Our replicated state machine utilizes lazy replication in order to simplify the replication process. In order to read and write to a particular state machine, frontends communicate with a leader which then manages the necessary replication and communication across the other replica managers.

2 Leader Election

We have implemented a very basic version of Paxos in order to facillitate leader election in the event that the leader goes down. The optimizations and simplifications we've made are as follows:

- All replicas are Learners, Acceptors, and Proposers.
- The proposing replica is also the distinguished Learner, and is tasked with forwarding the final leader to other replicas.

3 Load Balancing

Loads are distributed according to a very simple criteria. Each replica keeps a count of how many state machines it currently holds and the leader replica will distribute work accordingly.

4 Dataflow

Below are brief samples of the request response cycles that follow read, write, and creatoin operations.

- Client requests read of state machine s.
    1. Leader adds request to queue.
    2. Leader waits until all write requests to s have finished in queue.

3. Leader informs a replica manager a request is incoming.
4. Leader responds to client.
5. Client requests read from the correct replica manager.
6. Replica manager responds to client.

- Client requests write of state machine s.
  1. Leader adds request to queue.
  2. Leader waits until request is at top of queue.
  3. Leader asks all replica managers to write s.
- Client requests creation of state machine s.
  1. Leader asks all replica managers to create a new state machine s.