DS1 Project – Documentation

Code architecture:

1. The actors

Our simulation relies on 2 types of actors only: externalClient and Participant. The class Node from which Participant inherits is only a container made to extend the AbstractActor. However it still contains some useful getters and setters, variables as well as a few generic methods regarding delays, multicast and logging.

Practically, only one client object is needed throughout the simulation since it only transmits requests to the different nodes of the system. A number of N\_PARTICIPANT+1 (e.g. 6 in that particular case) Participant actors are created at the beginning. One is declared as coordinator and the others as simple participants. Each node has an integer id assigned, 0 for the initial coordinator and 1 to N\_PARTICIPANTS for the other nodes.

1. externalClient

This class only possesses a constructor, a Props method (necessary for the Akka actor creation) and one receive method for displaying the reception of read responses.

1. Participant

This class is the most important one : it contains all the methods for the coordinator and participant nodes. There are two constructors and two props methods for the initial actor creation, allowing to differentiate the creation of the first coordinator from the regular actors. Some useful getters and setters are declared below.

The bulk of the system relies on reception methods, divided into 3 behaviors :

1. Standard: This is the default operating mode, defining the way the system works during and epoch. It can process StartMessage, Update, WriteOK, UpdateRequest, ReadRequest, Timeout, CrashedNodeWarning, ElectionMessage, Acknowledgement, Heartbeat and CrashRequest classes.
2. Election : This receive behavior is triggered when the node in standard mode receives an election message. It is made to prevent the system from processing confusing updates during coordinator election. It is able to process : CrashedNodeWarning, ElectionMessage, ElectionAck, Synchronization and Timeout messages.
3. Crashed : When a node enters crashed state : it switches to this behavior that prevents it from processing any new message and from broadcasting anything.
4. The messages