Distributed Systems (CS60002)

Distributed Coordination Service (Group 06)

TA : Gulab Arora

Suyash Damle 15CS10057
Hardik Tharad 15CS10059
Anubhav Jain 15CS10062
Vivek Mudgal 15EE10055

Our Model

User Functionalities:

1. Add/Modify/Read/Delete File(params : dirPath, filename, …)
2. File Locking for series of updates
3. Delete server
4. Add server (any number of servers supported)

Internal Functionalities:

1. Leader Election
2. Heartbeat sending and recording (between all pairs)
3. Crash detection and recovery
4. 2-Ph commit protocol for various purposes

Model Assumptions

The following assumptions are made to simplify the model:-

1) Fully Connected Topology
2) Asynchronous and reliable system
3) FIFO channel
4) No link failure or Byzantine Fault
5) Client does not contact a failed server for read/write or make multiple attempts at various servers
6) Sponsor node does NOT fail while data is being copied-over to a new server
7) Not more than n-1 crash faults (n - number of nodes in the network)

System Guarantees

1. **Consistency Model:** Sequential Consistency
2. **Fault model:** Reliability, Availability: 100%, subject to fault model constraints

Specifications of Implementations (Brief Steps) :

1. Read : Serviced from local copies

2. Write / Modify: Write leader handles it. Standard 2-Phase commit involved.

3. Leader Election:
    a. Node that senses leader failure initiates (multiple initiators possible)
    b. Sends specific message to *all* others. As all nodes know about id and alive/dead state of all others, the present highest (or lowest) id node assumes responsibility of leader and informs all others.

4. Locking:
    a. Existing write leader initiates a 2-Phase commit to make everyone else agree that a series of writes is pending on a file.

5. Node addition:
    a. New node (M) sends out specific message requesting joining to the network
    b. First node to respond becomes *sponsor node (S)* for this node
    c. M gets leader IP and id from S. Contacts Leader Node (L) for id of itself
    d. L initiates a 2-Ph commit to come to an agreement on the id of the new node (max_id + 1 of all other nodes)
    e. M then copies over data and network info from S before taking request

6. Deletion (soft server removal):
    a. The client's request is sent to *all* nodes.
    b. The server servicing the client's request ensures agreement via a 2-Ph commit. Essential, as the node requested to be deleted might, at present, be acting as a leader or a sponsor or servicing a read. It would refuse temporarily.

7. Crash (hard removal):
    a. The node(s) detecting the fault via heartbeat absence or time-outs inform all others of the server having failed and update their own tables
    b. A server, on receiving such a message updates its own table

8. 2-Phase commit, wherever involved, would have safeguards against the leader failing and blocking all clients in the middle of the process.