

Assignment 8

Problem 1

If coordinator or subordinate node fails, then after restart we will first check the log to what's already written. If we have a "commit" or "abort" log record, but not an "end" record then we must redo/undo. If this is a coordinator then we keep sending "commit" or "abort" until we get acknowledgement from subordinate nodes. If we have a "prepare" log record, but not a "commit" or "abort", this node is a subordinate which means we have repeatedly contact the coordinator until we receive a status. If we don't have even a "prepare" log record, then we have to unilaterally abort and undo the Node.

Problem 2

(a) Example without restart:

T1	T2	T3
R		
	R	
	W	
		W

(b) Example with restart:

T1	T2	T3
R		
	R	
		R
		W
	W	

Problem 3

(a) $N=10, W=5$ $R + W > N$
 $R > 10 - 5$
 $R > 5$
 Minimum R is 6.

- (b) $R < N$
 $R < 10$
Minimum R is 1.

Problem 4

In Row 1 there no conflict because Copy 1 could be a descendent of Copy 2.

In Row 2 there is a conflict because of S_y and S_z . Both have a dependency issue that contradict each other.

Problem 5

In this graph there are thirteen nodes with two main loops. However, the loop on the left (A-F) are slowly going to be losing value because E shares its vote with F and M. M then passes this partial value into the loop on the right (G-L). This will cause the loop on the right to slowly collect all the value points over the course of iterations in the algorithm. This is why A-F and M will all slowly go to zero while G-L will continue to pass values around in a loop with a score of about $1/6$ for each node.

Problem 6

- 1) `db.restaurants.find({ "name" : "The Dead Rabbit" }, { "cuisine":1, "_id":0 })`
- 2) `db.restaurants.createIndex({ "name" : "text" })`
- 3) `db.restaurants.find({ $text: { $search: "Rabbit" } })`
- 4) `db.restaurants.aggregate([{ $group: { "_id": "$borough", "count": { $sum: 1 } } }])`