

CSE535: Distributed Systems

Phase 4 - Twins : BFT Systems Made Robust

Team Name: Server Side Squad

Vedhachala Tirupattur Shanmugam

Sai Bhargav Varanasi

Sujay Lakkimsetti

TEST REPORT

Config File: 'config/config.json' contains a list of all configurations to be executed.

Ledger File: For each configuration at index '\$c' mentioned in the configuration file, each validator with index '\$v' running scenario with index '\$s' creates its own ledger file under 'ledgers/config\$c/scenario\$s/validator_\$v.ledger'

Log File: For each configuration at index '\$c' mentioned in the configuration file, each validator with index '\$v' running scenario with index '\$s' creates its own log file under 'logs/config\$c/scenario\$s/validator_\$v.log'.

Explanation of all the variable names mentioned in the config file:

'nvalidators': Number of Validators/Replicas,

'nfaulty': Number of Faulty Validators,

'nclients': Number of Clients,

'nclientops': Number of operations each client performs,

'sleeptime': Delay between consecutive operations for the same client in seconds,

'clienttimeout': Amount of time the client waits in seconds to receive the response. If no response is received, it retransmits that request

'delta': Amount of time in seconds used to decide the pacemaker timer timeout time.

'window_size': Window size used for Leader Election,

'exclude_size': Exclude size used for Leader Election,

'twin' : An object pointing faulty validator_id to it's twin's id

'nocommit_threshold' :

'max_idle_round_timeouts' :

'n_partitions' : Number of partitions to divide the given list of validators and it's twins if any

'max_rounds' : Enumeration limit on the number of rounds generated for each testcase

'partition_limit' : Enumeration limit on the partitions_list generated
'partition_leader_limit' : Enumeration limit on the partitions_leader_limit being generated
'max_testcases' : Enumeration limit on the max number of testcases generated for each configuration
'seed': To generated pseudorandom numbers
'is_Faulty_Leader' : Boolean flag indicating if only non-honest validators can be leaders
'is_Deterministic' : Boolean flag indicating deterministic or randomised testcase generation.
'N_fail_type_limit' : Enumeration limit for type-based message loss.

Fixes made to DiemBFT:

1. Sync up lagging replicas
2. Mempool issue resolution to remove committed transactions.

Testcases:

Case 1 : To show the working of scenario generation and network playground (used to intercept messages).

Refer to 'src/testcases_report.json' case1:

Explanation: Scenario generator generated this testcase and the network playground is able to intercept messages and deliver messages to intended nodes based on partition-set information. All the validators are able to commit transactions received from client as indicated from the network playground's log.

Case 2: To show syncup of a slow validator when it receives a proposal message after staying outdated for several rounds.

Refer to 'src/testcases_report.json' case2:

Explanation: Through this testcase we were able to test our implementation of syncup of slow validators. Generated testcase has put validator 4 into a non-quorum partition till the end of round 9. In round 10, validator 4 receives a proposal message from leader 2 when it requests for syncup of missing blocks and gets synced up when 2 sends blocks comparing the high_qc of 2 with high_qc of validator 4.

Validator requests for blocks from the proposer. After receiving the response for syncup, Validators 4 process all the qc for safety and gets updated.

Case 3: To show the liveness failure while executing DiemBFT using twin's scenario executor.

Refer to 'src/testcases_report.json' case3:

Explanation: Liveness failure is identified when only TCs are formed for a threshold of rounds(5 by default) or if the majority of nodes are stuck on different rounds (TC won't get formed).

In this example, Liveness is violated as no commit happened in majority of nodes for 5 rounds.

Case 4: Injected bug changing quorum from $2f+1$ to $2f$

Refer to 'src/testcases_report.json' case4:

Explanation: Took time less than 1 sec to see out of order commits and lead to safety violation.

Case 5: Injected bug changing quorum from $2f+1$ to $2f+2$

Refer to 'src/testcases_report.json' case4:

Explanation: Majority of the test cases have failed with liveness violation(no commit for 5 rounds)

Case 6: Injected bug by changing high_Qc to always point to qc in blocktree.process_qc method

1. By changing `high_qc <- qc` always instead of performing a
 - a. `Self.high_qc = get_qc_with_max_round(qc, high_qc)` check, I see all blocks getting successfully committed for several scenarios. (1st and 2nd test case generated using config1).

