CSE535: Distributed Systems

Project: DiemBFT v4 Consensus Algorithm Phase 2

Team Name: Loyal Byzantine Generals

Vivek Neppalli

Manish Adkar

Shubham Sahu

User Manual

- **DistAlgo Installation**
  - Follow https://github.com/DistAlgo/distalgo/ to see the steps to install DistAlgo in the system.
- **Known system-specific issues**
  - For macOS, If the message size exceeds the threshold specified by the system for UDP, the system will throw an error *"OSError(40, 'Message too long') ".* Execute `sudo sysctl -w net.inet.udp.maxdgram=65535` to increase the limit.
- **Configuration and output for diembft simulation**

  All the test cases follow the same format for the config file; ledger files for validators; and log files for all validators as well as clients mentioned in the following format.

  Config File:

  '*config/config.da'* 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'* creates its own ledger file under '*ledgers/config$c/validator_$v.ledger'*

  Log File:

  For each configuration at index *'$c'* mentioned in the configuration file, each validator with index *'$v'* creates its own log file under '*logs/config$c/validator_$v.log'.*

For each configuration at index *‘$c‘* mentioned in the configuration file, each client with index *‘$r’* creates its own log file under ‘*logs/config$c/client_$r.log’*

Example of configuration file located at *“/config/config.da”*

Keep the import statements and only modify the existing configs list.

Each element of the configs list is the configuration for the simulation.

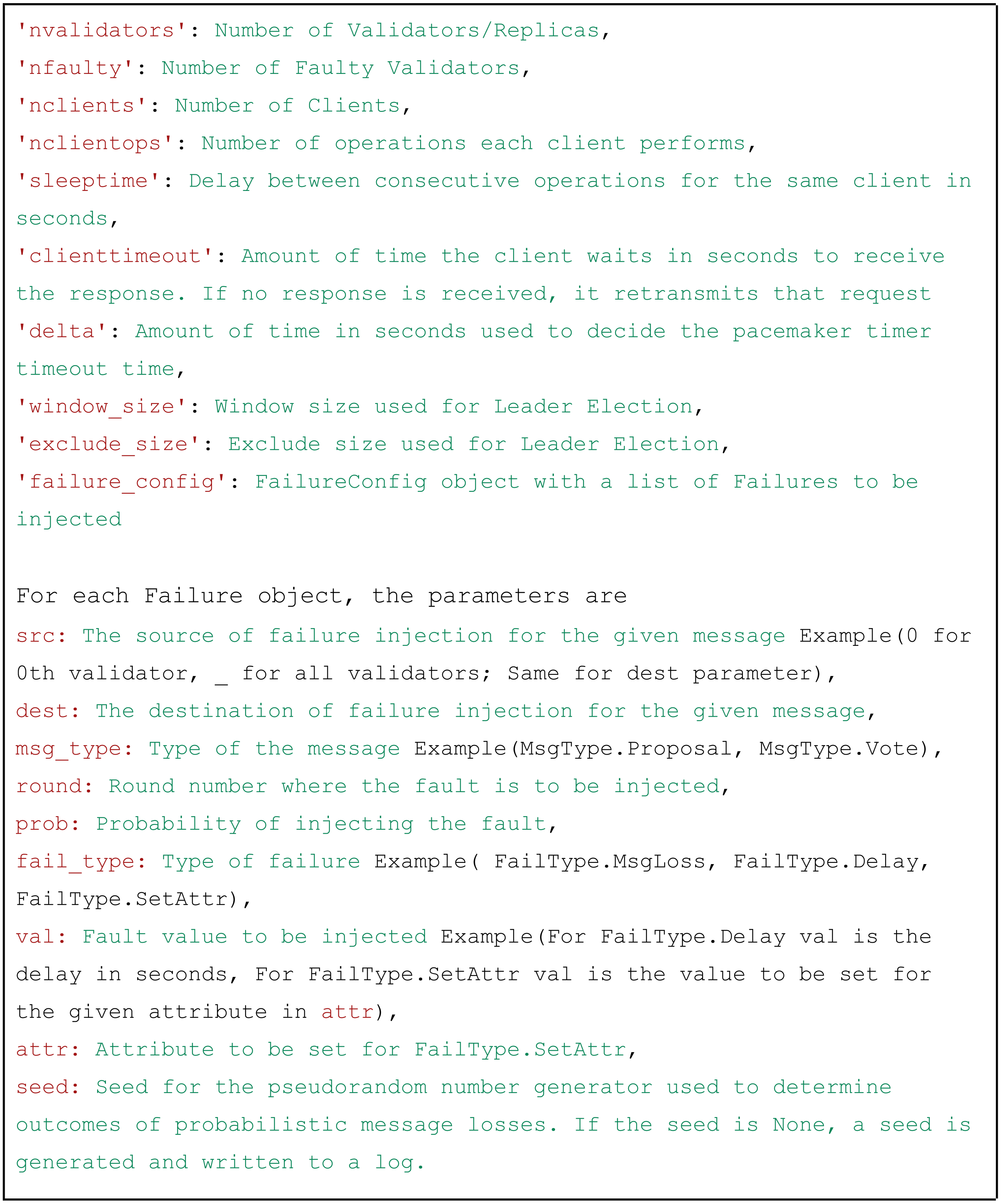
The simulation runs all configurations mentioned in the configs list in a single execution

```
from object_types import FailType, Failure, FailureConfig, MsgType

configs = [
{
   'nvalidators': 5,
   'nfaulty': 1,
   'nclients': 5,
   'nclientops': 5,
   'sleeptime': 1,
   'clienttimeout': 10,
   'delta': 5,
   'window_size': 5,
   'exclude_size': 1,
   'failure_config': FailureConfig(
       failures=[
           Failure(src='_', dest='leader', msg_type=MsgType.Vote,
                   round=3, prob=1, fail_type=FailType.MsgLoss,
val=None, attr=None),
       ],
       seed=12345678
   )
}]
```

Figure 1. Example of configuration file located at *“/config/config.da”*

- **Explanation of each label present in the configuration 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,
'failure_config': FailureConfig object with a list of Failures to be
injected

For each Failure object, the parameters are
src: The source of failure injection for the given message Example(0 for
0th validator, _ for all validators; Same for dest parameter),
dest: The destination of failure injection for the given message,
msg_type: Type of the message Example(MsgType.Proposal, MsgType.Vote),
round: Round number where the fault is to be injected,
prob: Probability of injecting the fault,
fail_type: Type of failure Example( FailType.MsgLoss, FailType.Delay,
FailType.SetAttr),
val: Fault value to be injected Example(For FailType.Delay val is the
delay in seconds, For FailType.SetAttr val is the value to be set for
the given attribute in attr),
attr: Attribute to be set for FailType.SetAttr,
seed: Seed for the pseudorandom number generator used to determine
outcomes of probabilistic message losses. If the seed is None, a seed is
generated and written to a log.
```

Figure 2. Explanation of each label present in the configuration file

- Commands to execute in sequence

  -
    ```
    cd <path_of_project_folder>/src
    ```

  -
    ```
    python3 -m da --message-buffer-size 65535 run_diembft.da
    ```