ViewStamped Replication - Revisite

Problem Description

There are numerous algorithms out there which try to solve the problem of node replication (state machine replication) correctly and efficiently. It is however important to justify why these algorithms are correct. This requires thorough reasoning and testing to make sure that the algorithm is robust and will achieve the desired properties even in case of failures upto some known limit. In this project we will be studying the ViewStamped Replication algorithm described in the paper ​***“ViewStamped Replication Revisited” by Barbara Liskov and James Cowling[2].*** ​​We have implemented the algorithm from scratch using the pseudo-code described in the paper. We have also tested the algorithm for correctness and generated different performance graphs and analyzed them.

Implementation Files:

|  |  |
| --- | --- |
| File Name | Functionality |
| vr\_revisited\_new.da | Main Viewstamped Replication Implementation |
| Monitor.da | Monitor program |
| driver.py | Driver program and error injector program |
| test.cfg | Configuration file for testing |

Input Parameters :

Our code takes the following parameters as input:

|  |  |
| --- | --- |
| INPUT PARAMETERS | EXPLANATION |
| f | Number of max replica failures supported, This sets the number of replicas to 2f+1 automatically |
| c | Number of clients |
| req | Total number of requests per client |
| c\_tout | Client timeout |
| r\_tout | Replica timeout |

Running the programs:

There are two modes in which the user can run the program:

Execution Mode 1:

This mode is the one where the user just wants to run the algorithm for a particular set of execution parameters as described above in the table. To run the program in this mode, give the following command in the source directory:

**Command: python -m da vr\_revisited\_new.da f c req c\_tout r\_tout**

**Output**​​: In this mode the output will be available as **logs** from the system on the console.

Execution Mode 2:

This is an automated testing mode where the user just needs to run the driver program and then the driver program takes the job of giving parameters to the algorithm and also takes care of generating correctness and performance statistics.

Run the following command in the source directory:

**Command:** ​​Python driver.py

**Output**​​: In this mode the driver program will run the algorithm for multiple parameter values and in the end will output the correctness and performance statistics in the “​**output\_vr\_rev**​​” folder as ​**“validation.csv”.** ​​It will also generate the performance graph.

Note: In order to change the variable against which the performance statistics should be obtained, one need to change manually in the driver program code to produce graphs.