Parallel Execution of Automated Test Scripts

Ashwini N S
MTech (Software Engineering)
Information Science Department
RVCE Bangalore, Karnataka, India

B.K Srinivas
Assistant Professor
Information Science Department
RVCE
Bangalore, Karnataka, India

Ganesh Arunachalam
Member Technical Staff
CPE-DPG-QA
NetApp
Bangalore, Karnataka, India

Abstract—Testing is very important process of checking software, whether it fulfills customer needs according to their requirement. Approximately 40% of the time taken to deliver the product tends to be spent on testing. Objective of this paper is to achieve more than 50% reduction in testing time for all the automated tests that can be run in parallel.

To achieve this objective a driver framework is designed which supports execution of multiple test scripts in parallel, with some constraints.

Constraint satisfied test scripts when executed in parallel saves 80% of execution time with respect to serial execution, for the same set of scripts and also reduces the number of hardware resources needed for testing.

Keywords—Driver framework, Customer needs, Parallel execution of scripts, reduces number of hardwares required.

I. INTRODUCTION

Every software product needs to be tested for its quality and functionality before it is delivered to customer. Testing process consumes approximately 40% of software development lifecycle time [1][2][3]. If product testing time is reduced by half, the time saved, allows for a shorter cadence in the product delivery to customer / allow for fixing bugs earlier by finding them earlier in the cycle.

While Automation has reduced testing time and effort compared to manual runs, existing testing process follows serial execution of scripts. These sequential testing mean, if a test is identified to be run late in the cycle and it leads to the identification of a bug, then fixing the bug and testing the fix as well as running additional regressions for the fix is delayed and causes a lot of angst in the system. After fixing the current bug, other functionality of the software might get affected which need to be tested again (Regression Testing). In case of serial execution of scripts, the time available nearer the end of the cycle to handle such bugs is limited and may result in a field escape if the choice of regression tests are not handled effectively and if all regression tests need to be run it may lead to a delivery time slippage. To avoid this situation time needed to run a large part of the regression suite must to be reduced.

To reduce the testing time, Test scripts need to be executed in parallel by designing a framework which supports parallel execution of test scripts. All Test scripts cannot be executed in parallel since some test scripts depend on output of other scripts; others affect the functionality of the hardware. If test scripts need to be executed in parallel, several constraints need to be satisfied, ex:- there should not any dependency among scripts, scripts should not affect the functionality of the hardware, variables of the
scripts must be independent with respect to any global variables.

In this paper Section 2 describes the Existing Architecture for serial execution of scripts. Section 3 describes proposed architecture for serial execution of scripts. Section 4 describes Algorithm used for development of driver framework. Section 5 describes Algorithm used for selection and grouping of test scripts in driver. Section 6 describes Implementation process. Section 7 describes Comparison between existing and proposed results. Section 8 describes the Conclusion and Future work.

II. SYSTEM ARCHITECTURE FOR SERIAL EXECUTION OF SCRIPTS

Setup the environment by specifying the Automated Environment Tool host files which contains host-type, connection authentications, partner node details etc., give the necessary parameters which include job list i.e. set of tests to be executed in serial, pass these to any automated testing environment, which executes the tests in serial by accepting necessary parameters for successful execution of test script from the plan file, store the results of the execution in logs as shown in figure 2.1

Disadvantage:-
- Time consuming Activity.

III. SYSTEM ARCHITECTURE FOR PARALLEL EXECUTION OF SCRIPTS

Setup the environment by specifying the Automated Environment Tool host files which contains host-type, connection authentications, partner node details etc., provide the job list which has test scripts to be executed in parallel, pass test scripts to any automated testing environment, which executes the tests in parallel[4][5] by accepting necessary parameters for successful execution of each test script from the plan file. Store the results of the execution in logs as shown in figure 3.1

Driver manager in turn executes init (which accepts all the necessary parameters from command line) and setup (which creates necessary aggregates, volumes, block mirror relationship, vault relationships etc). After its successful completion return the control to supervisor and execute client setup (which finds the necessary aggregates, volumes, interfaces etc…) and co-job after its completion driver supervisor transfers the control to the worker of driver framework where each and every worker executes individual tests (like performing volume move and backup in parallel, performing backup and restore of volumes, performing remote volume move from one aggregate to other of different filer, performing local volume move from one aggregate to other aggregate of different nodes of same filer) in parallel, once all the tests are executed, return the result to driver supervisor which in turn returns the result to the driver manager, while executing the test scripts by interacting with the clients, switches and filers.

Logdir
ATE libs
Filer_list
Parameters
Pretest, posttest, Cojob, coreset, prejob

env_setup
Automated Test Environment (ATE)
Job list
Pretest, posttest
Prejob

ATE, STAF, ACL (Automated Common Language) and User Interface

Host files
Storage appliance
Windows or Linux machine

Figure 2.1

Ashwini N S, B.K Srinivas, Ganesh Arunachalam
IV. ALGORITHM USED FOR DEVELOPMENT OF DRIVER FRAMEWORK

1. Develop a manager module.
   1. start
   2. Count number of clients
   3. call those many threads of supervisor module.
   4. wait for result from all supervisor module.
   5. print result.
   6. stop

2. Develop supervisor module
   1. start
   2. Count the number of test scripts for parallel execution.
   3. call those many threads of worker module.
   4. wait for result from all worker module.
   5. pass the result to manager module.
   6. stop

3. Develop worker module
   1. start

2. execute the script.
3. pass the result to supervisor module.
4. stop

V. ALGORITHM FOR TEST SCRIPTS SELECTION FOR PARRAL EXECUTION

Algorithm for selection of test scripts and grouping them in drivers for parallel execution

1. Select the independent test scripts.
2. Test scripts selected for parallel execution should not affect hardware used by all scripts.
3. Modify the scripts to resolve conflicts without affecting its original functionality.
4. Collect the execution time of the test scripts
5. Group the 8 to 10 test scripts which consumes approximately same time for its execution.
6. Repeat step 4 and 5 until all test scripts are grouped in drivers for parallel execution.

VI. IMPLEMENTATION

Create all the necessary environmental setup needed to execute scripts. Accept the test to be executed from the job list. Provide it to Automated test environment (tool for executing scripts). Then driver mangerframework counts the number of clients on which scripts to be executed and those many supervisors to take care of parallel execution gets created. Supervisor counts the number of test cases that individual driver has so those many scripts starts the execution at a time. Each Driver is executed serially but test scripts inside it are getting executed in parallel.
VII. COMPARISON BETWEEN SERIAL AND PARALLEL EXECUTION OF SCRIPTS

Time taken for each and every script depends on the area of the script; these are the execution details of the scripts in tape area.

Again same set of test scripts are executed in serial by fixing the test bed with all the requirements necessary for successful execution of scripts and little changes made to script to improve the efficiency.

By comparing serial execution of scripts Before Fixing (Serial execution BF) and parallel execution 80% execution time of test scripts are saved.
VIII. CONCLUSIONS AND FUTURE ENHANCEMENT

The PEAT reduces the number of resources required to test the software, it also reduces the time required to execute the script, so that bugs are detected in early stage. Depending on availability of hardware each driver execution can be done in parallel.

REFERENCES


