Regression Test List Sharding in a Distributed Test Environment

[1,*]Michelle Gonsalves, [1]Jyothi Mandala
[1] Dept. of Computer Science and Engineering, Christ (Deemed to be University), Bengaluru,

Abstract— One of the major issues during the regression test of the new version of Real Time Operating System (RTOS) is the time involved in test case execution. The main reason being a single embedded system device under test (DUT) is used to execute the test list containing several test cases. This traditional method of regression test also leads to wasted productivity of the other devices at hand that could be otherwise used during this regression test. Hence, in this paper, we propose a technique that aims at reducing the overall regression test cycle time of a newer version of a Real Time Operating System (RTOS) by employing a method known as "test-list sharding" in a distributed test environment. In the proposed work, multiple DUTs are connected to the test server via a communication network. The test server executes the test list containing several test cases and performs the test-list sharding, that is, distributing test cases to different DUTs and executing them in parallel. After the test is executed on the DUT, the test results are sent back to the test server which will summarize all the results. In the proposed work, the sharding is done by distributing the test cases without overloading or under loading any of the DUTs. Test list is sharded in such a way that the same tests are not sent to multiple DUTs. The main advantage of the proposed method is that the test sharding can be easily scalable to accommodate any number of devices that can be connected to the test server. Also, the test list sharding is done in a dynamic way so that the tests are distributed to an idle DUT that has completed a test execution and ready for another test to execute. The comparison study of executing a sample test list sequentially on a single DUT and distributed test system with multiple DUTs is performed. Results obtained showed the performance gain in terms of test cycle time reduction, scalability, equal load distribution and effective resource utilization.

Index Terms— Real Time Operating System, Test-list Sharding, Embedded system, Regression Test, Distributed Test Environment.

* Corresponding Authour: michelle.gonsalves@mtech.christuniversity.in

Make sustainable green world by using Renewable energy

Arifa Parvin[1,*], Adeeb Ahnaf[2]

[1]*Electromedical, Dhaka Mohila Polytechnic Institute, Bangladesh*
[2]*Architechture, Bangladesh University of Engineering and Technology, Bangladesh*

Abstract:Renewable energy is key to the solution of a safer, cleaner, and sustainable for green world. It comes from natural resources that are sufficient and continuously recharged.Renewable energy sources which are achievable in sufficient all around us, provided by the sun, wind, water, waste, and heat from the Earth are restored by nature and transmit no pollutants into the air.Renewable energy sources includes with safety environment, health impacts of sustainable for green world.Solar energy converts into usable electric energy. But when operating solar energy do not produce air pollution.Sothe technology are used for heat, lighting, hot water, electricity and even cooling for homes, businesses and industries At last electric vehicles are reduce to carbon footprint in their own way.So this provision will provide a clean environment to achieve renewable energy targets for future generations.

Keywords:Sustainable, green world, future generation.

Introduction

Necessity of energy demand is required for every society with a view to meeting the basic needs. Also we are now entering Electric Vehicle (EV) sector which will increase the demand of electricity.We Know the main source of electricity are coal and natural gas which materials are decreases naturally.So this provision can help meet the dual goal of reducing our greenhouse gas emissions by shifting the energy sector to renewable energy.

We know that renewable energy is always continuously replenished by nature. For our developing countries, this is a golden opportunity, because it replaces environmentally harmful methods of energy production. This energy sources includes such as sunlight, wind, rain, tides, waves and geothermal heat etc.

Environmental damage is now present issue. This technology improves global environmental issues, energy health and climate impacts. At the same time, this technology affects the distribution of the economy using natural resources.

Sustainable development is not possible without standards.Here we discuss the sustainable development using the case of standardization in the field of energy efficiency.So the uses of renewable energies in city scale the consumptions of non-renewable resources can be reduces which will put the next step towards are zero carbon emission city.

* Corresponding Authour:arifadmpi@gmail.com

With the advent of Real time Open Source Operating systems that support different types of embedded systems devices, programmers have started focussing only on developing and testing applications assuming the correctness of the underlying operating system. Thus, whenever a new version of such an Operating System (OS) is released, it is very important to test it on all the supported embedded system devices. A comprehensive OS test list would generally have a huge list of test cases for features such as the User Interface, device drivers supported, playback of media and various other functional tests. Regression testing is an essential task that is performed before releasing a new version of software to gauge its performance and software version compatibility. In case of Real Time Operating System (RTOS), this is done to verify the performance of the new version of the operating system as expected with all the supported Embedded System Devices. These embedded devices are called the Devices Under Test (DUT). To test the RTOS version's compatibility, a test list for a particular type of DUT is executed. Each test list contains several test cases which are present on a centralized test server, and the supported DUT is connected to and communicates with the test server via a communicating network. When the test list is scheduled for execution by the test server, each of the test cases is sent to the DUT and gets executed. After each test case completes execution, the test results are obtained from the DUT and saved on the test server for further analysis. Since, test results are obtained one by one till all the testcases in the test list are executed on the DUT, it usually requires a significant amount of time to complete the execution of the given test list. Also, this method utilizes only one of the many DUTs that might be available with the test team at hand during this testing and regression cycle. Hence, this method leads to wasted productivity of the other devices that could be used for testing.

To reduce regression testing time and to utilize existing DUTs, in this paper we propose a regression test list sharding technique. A distributed test environment is created by using multiple DUTs of the same type. Test cases are sharded across DUTs thereby reducing the overall regression testing time. The rest of the paper is organized as follows. Section II provides literature review of existing work. Section III presents the experimental setup and section IV provides results and analysis. Conclusion and future work is described in Section V.

LITERATURE SURVEY

With the surge in mobile applications development, their diverse features and platforms, along with the increased complexity in mobile app testing and quality assurance has become very crucial. Also, there is coexistence of several versions of the OS available and operational at any given point in time. Currently, several researchers are working towards solving issues related to coexistence of different OS versions and Device versions. Regression testing is one of the very important tests which is used to check whether the newly released software has not affected already existing features [1].

Regression can be applied at different levels such as unit, integration or system level testing [2]. Regression testing also ensures compatibility of newer versions of the software with the older versions. Several researchers proposed different techniques to reduce the cost of regression. Three important factors that influence the cost of regression testing are test selection, test suite minimization and test case prioritization [3]

Do et al. [4] proposed a new technique to choose regression test for Android apps. The approach used a combination of static impact analysis and dynamic coverage information. DetReduce tool was proposed by Choi et al. [5] which is used for Android GUI testing.

Liu et al. [6] proposed a system to test android device compatibility with the Android compatibility standards. To reduce the testing time, the proposed method decomposes the test list tasks and is allocated to android devices to execute concurrently.

X. Bai et al. [7] made a survey of representative approaches and typical tools for cloud testing. It identifies the needs for cloud testing tools including multi-layer testing, SLA-based testing, large scale simulation, and on-demand test environment.
Tao Zhang et al. [8] proposed one optimized compatibility testing strategy using a statistical approach to reduce test costs and improve engineer's operation efficiency. A compatibility testing service has been proposed for mobile apps. Two case study results are reported to demonstrate its potential application and effectiveness.
Naith, Q. and Ciravegna, F [9] have proposed a crowdsourcing testing strategy for mobile testing. In this system, the general concept of crowdsourcing is dividing big projects into small tasks that can be easily achieved by crowds. This paper had an interesting strategy where it announced the testing tasks to crowd testers, by converting the submitted project to multiple tasks to be easily performed by the crowd tester. This enables testing to be achieved faster. However one should ensure that no tests are missed while breaking down the tasks.
In this paper, a new version of RTOS is subjected to regression testing using a technique called test list sharding. Proposed technique uses dynamic allocation of test cases to various embedded devices thereby reducing the execution time.

EXPERIMENTAL SET UP

Fig1 shows the regression testing experimental setup based on test list sharding. It consists of a test server connected to multiple DUTs of the same type via a communication network. DUT can be any device such as Raspberry Pi, TV, Arduino Development board, etc. Test cases are populated in a queue which is maintained by the test server. A test list file and device information file are the two inputs to the test list sharding tool. The main test process creates a test thread for each DUT which is responsible for delivery of test cases and collecting the result logs from the respective DUT. To ensure that two DUTs do not pick the same test case, a lock is provided to the test queue. Test queue access through a thread lock will ensure mutual exclusion of the test cases. When a DUT completes the current test case in hand, it can be allotted with a new test case. This ensures the efficient device utilization, load balancing and dynamic sharding of the test list. When the testing is completed, that is, when the test queue is empty, the threads should rejoin the main test thread. The main test process then creates a single Local HTML report to summarize the test results. Also, an online HTML test report will be generated to summarize the test results of the test list.

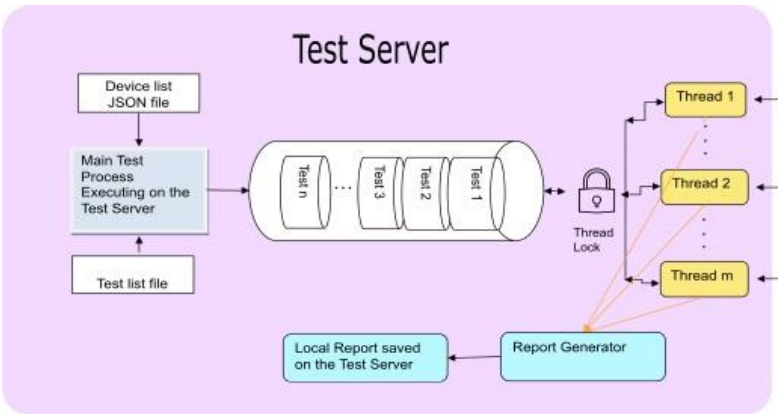

Fig 1. Regression testing experimental setup.

RESULTS AND ANALYSIS

In the current experimental setup, a Linux based system is used as a test server and two Raspberry pi are deployed as DUTs. Test sharding tool is implemented in the Python scripting language. Test queue had 19 sample test cases. Fig 2 and Fig 3 shows the sample HTML report generated with a single DUT and two DUTs respectively. When the test list ran on a single DUT and two DUTs, the execution time was around 6.46 min and 3.29 min respectively. Hence, there is performance gain of around 51%. The proposed system can easily be scaled to accommodate multiple DUTs since it requires creating a new thread for each additional DUT.

CONCLUSION

The paper presents a regression test based on test list sharding. The main idea behind the approach is to dynamically allocate test cases to DUTs of the same type in a mutually exclusive manner. The approach offers several advantages such as reduced regression test cycle time, effective load balancing among DUTs, and easy scalability. The experimental results indicated that there is significant performance gain of 51% in terms of reduction in execution time.

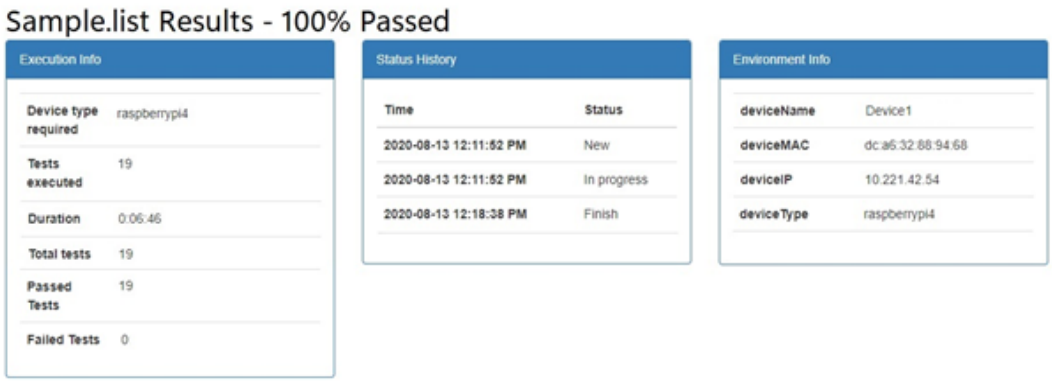

Fig 2. Test Sharding Tool results with a single DUT

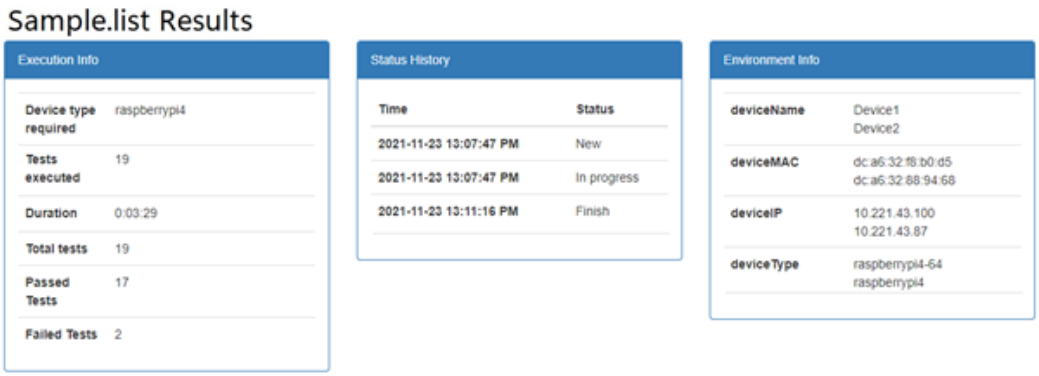

Fig 3. Test Sharding Tool results with a two DUTs

REFERENCES

[1] IEEE.IEEE Standard 610.12 IEEE Standard Glossary of Software. 1990. .

[2] IEEE Standard for Software Maintenance. IEEE Std 1219-1998. Oct 1998

[3] G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold. “Prioritizing test cases for regression testing,” IEEE Transactions on Software Engineering, vol. 27, no. 10, pp. 929–948, 2001

[4] Do, Q., Yang, G., Che, M., Hui, D., & Ridgeway, J. (2016). Regression test selection for Android applications. Proceedings of the International Workshop on Mobile Software Engineering and Systems - MOBILESoft ’16.doi:10.1145/2897073.2897127

[5] Choi, w., Sen, K., Necula, G., Wang, w. (2018). DetReduce: Minimizing Android GUI Test Suites for Regression Testing. ICSE ’18, May 27-June 3, 2018, Gothenburg, Sweden

[6] C. -H. Liu, W. -K. Chen and S. -L. Chen, "A Concurrent Approach for Improving the Efficiency of Android CTS Testing," 2016 International Computer Symposium (ICS), 2016, pp. 611-615, doi: 10.1109/ICS.2016.0125

[7] X. Bai, M. Li, B. Chen, W. Tsai and J. Gao, "Cloud testing tools," Proceedings of 2011 IEEE 6th International Symposium on Service Oriented System (SOSE), 2011, pp. 1-12, doi: 10.1109/SOSE.2011.6139087.

[8] Tao Zhang, Jerry Gao, Jing Cheng, and Tadahiro Uehara, “Compatibility Testing Service for Mobile Applications,” Proc. of the 2015 IEEE Symposium on Service-Oriented System Engineering (SOSE '15), Washington, DC, USA, 2015, pp. 179- 186.

[9] Naith, Q. and Ciravegna, F. (2018), "Mobile devices compatibility testing strategy via crowdsourcing," International Journal of Crowd Science, Vol. 2 No. 3, pp. 225-246.