ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

A Survey on Regression Test Selection Techniques

Aditya Akotkar, Prof. M. S. Wakode

M.E Student, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

Asst. Professor, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

ABSTRACT: Regression testing is an expensive testing procedure utilized to validate modified software. Tester struggles to selectively run relevant tests for pre-testing defects in software. Standard set of tests identified based on features may not include all the impacted tests for pretesting a fix made at different layer. Also, it is inefficient to execute all the tests for a small change in code. Thus to reduce the cost of testing and to improve the effectiveness, there is a need of identifying, selecting and executing impacted tests based on changes in the code.

The aim is to develop a utility to select and execute optimal and relevant number of tests that would provide maximum test coverage with minimal number of tests for C++ based applications. The idea is to create a database to map the functional tests and C++ code files by collecting coverage data and then grouping tests based on multiple techniques. Finally, integrating this utility into existing testing process for selecting tests based on changes in code.

KEYWORDS: Code Coverage, Regression Testing, Regression Test Selection.

I. INTRODUCTION

Software systems continuously evolve during development and maintenance. A software is changed for a variety of reasons, such as fixing defects, adding new functionality, and improving performance. It is important to check new functionality as well as old functionality is still working correctly. After a software is changed, the modified version of the software should behaves as intended, and that modifications should not adversely impact its quality. This is done by regression testing. General approach of regression testing is to test all the test cases. Thus regression testing is expensive. It takes up to 80 percent of testing budget and 50 percent of maintenance cost. Regression testing may waste resources such as tester's time and computation resources.

Till date various prioritization methods are proposed. Prioritization helps in early detection of defect but it does not reduce cost of regression testing. Regression test selection is a well-known problem in software testing. Regression test selection techniques select smaller subset of large regression test suite for testing. 'Retest all' approach is not suitable for large test suites as resource requirement is very high. Also it is inefficient to execute all tests for small change in code. Thus it is necessary to trim the test suite.

II. RELATED WORK

Researchers have explored and proposed various methods for reducing cost of regression testing by applying various test selection and prioritization methods. Survey of such methods is carried out by Rothermel [13] and many others [11][8][6].Though large number of methods are proposed, regression testing in industry is still using conventional approach i.e. test all approach. Researchers have given different names to different techniques, but the core logic is similar. These techniques can be broadly classified as follows.

1. By Line

This is very basic test case selection method based on code coverage [14]. In this approach only that test cases are selected which traverses through modified source lines. In practical, considering only modified lines may not give impacted test as change in one line may affect others. Also it is hard to determine actual modified line numbers. To

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

solve this problem, selection of lines is normalized by considering code block of five lines (above two lines and below two lines) and test cases which pass through this block are selected. Tests selected by this approach are large in number.

Advantages of this approach is that only targeted tests are selected that execute code around the changed line. Also, it gives fewer tests than all tests.

Disadvantage of 'By Line' is that test suite may be too large to run in available time. It is possible that 'By Line' may return no test in case of new code.

2. Max Min

This is a statistical approach to solve regression test selection problem. In this approach minimum number of tests are selected such that their combined coverage is maximum. This technique ensures that most of the code is tested. Tests selected by this method are less than that of By Line. Similar technique is used in various prioritization algorithms. 'Max Min' do not consider modified line numbers for selecting tests, instead file level granularity is selected. Survey of such techniques is given by Rothermel [5].

Advantage of 'Max Min' is that it consistently provides minimum number of tests that execute changed source file. Disadvantage of this method is that, test that would find a problem are often excluded. This is least reliable method.

3. Intersect

This is hybrid approach of solving regression test selection problem. This method is combination of By Line and Max Min. Tests which traverse through modified part and provides high coverage are selected. Kandil [1] proposed similar approach which is combination of test case selection and prioritization. As tests are selected by combining above two methods, tests selected are less in number. Disadvantage of this method is same as 'Max Min', as it may exclude impacted test.

4. Clustering

In addition to above techniques clustering technique can be used in test case prioritization [3] [4]. In clustering technique, test cases are clustered based on certain properties such as code coverage, previous fault history etc. Clustering groups similar test cases in one group and dissimilar test cases in different group. It is an unsupervised learning approach as class labels are not known in advance. Different techniques ensures that dissimilar tests are selected so as to provide high fault detection capability.

By Line technique works on approximation, which is not very reliable in case of coding. By selecting function level granularity, we get test cases traversing through function which contains modified code. By using function level granularity, we get more number of tests than line level granularity. Combination of 'By Line' and 'By Function' will select modification revealing test cases. It will not select tests outside function boundary which can be selected by 'By Line'. Combined approach will select large number of redundant tests which can be removed by clustering them on the basis code coverage.

III. SYSTEM ARCHITECTURE

Figure 1 describes COVERAGE DB architecture. It provides automatic selection of impacted test. Customized code coverage reports are provided to tester so that reports can be analysed to add new functional tests. System provides grouping of tests by multiple techniques (By Line, Max Min, Intersect and Proposed new technique).

Automated tests are executed on code under test. With the help of coverage tool, code coverage is gathered. As code under test is C++ code, we have used OpenCppCoverage tool for gathering code coverage. OpenCppCoverage is free and open source tool for exe applications. Customized coverage report of test suite is generated. Gathered code coverage data is stored in database. Database stores mapping of test and source code at different granularity (file, function, line). While providing defect for retesting, developer needs to provide source files. After getting modified file information from defect system, COVERAGE DB compares modified file with original file with the help of version control software. Now the source code lines is mapped to test cases and test cases are fetched by using techniques in the literature as well as proposed technique

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

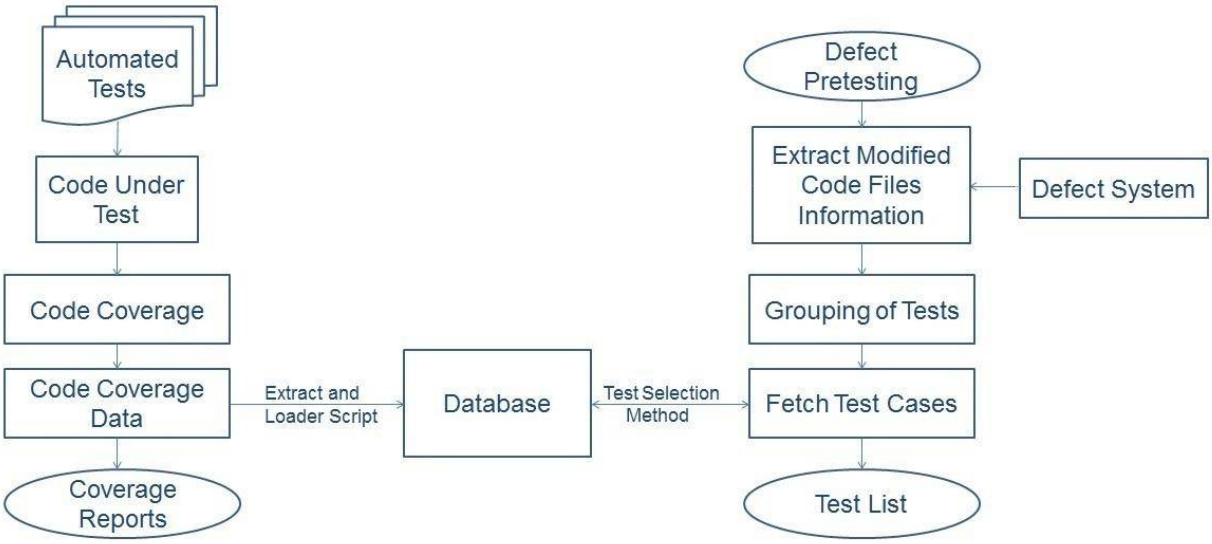

Fig. 1. System Architecture

IV. PROPOSED METHOD

Figure 2 describes proposed test case selection technique. It combines 'By Line' and 'By Function' method to select modification revealing tests. For modified line which is at function edge, 'By Line' approach may select lines outside the function and thus ineffective tests. Combination of techniques will discard such ineffective tests.

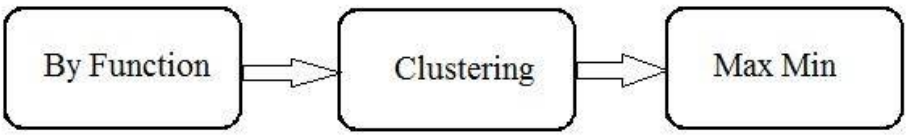

Fig. 1. Proposed test selection method

Algorithm

1. Execute test suite on the code under test.
2. Generate code coverage report at line level and function level granularity.
3. Select regression test suite based on 'By Line' method.
4. Trim obtained test suite by selecting matched test with 'By Function' method.
5. Apply clustering technique on trimmed test suite to remove redundant tests.
6. Apply 'Max Min' to get final reduced test suite.

Clustering is done on the basis of code coverage. Output of clustering is groups of similar test cases. These similar test cases can be a same test case with different arguments for checking boundary conditions or different test cases checking same code. Such redundant test cases are removed by clustering. Last step selects test case from each cluster which has maximum coverage.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

V. CONCLUSION

In this paper we discussed use of clustering technique in regression test selection and reducing testing cost. We also proposed combined technique of 'By Function' and 'clustering' method to give advantages of both. Proposed technique overcomes drawbacks of byline technique and provides effective test set.

Proposed technique is combination of test case selection and prioritization technique offering advantages of both. It is suitable for large scale systems with large test suite. Proposed technique reduces the cost of regression testing.

VI. ACKNOWLEDGEMENT

This work is carried out as an intern project in SAS Research and Development. Pune. I take this opportunity to express my deep sense of gratitude towards Mr. Kishore Jain, Manager, Quest, SAS Research and Development Pune, for giving me this splendid project for my dissertation work. I would like to thank Mr. Atul Kachare, Tech Lead, SAS Research and Development Pune, for his valuable guidance. I would like to thank my project guide Prof. M. S. Wakode for her inspiration, priceless suggestions and support for this project. I wish to express my thanks to Dr. Rajesh Ingle, HOD Computer Department, PICT, for encouragement and providing best facilities. I thank all the staff members for their indispensable support and for most valuable time lent as and when required. I thank all the people who are directly or indirectly involved in this project.

REFERENCES

1. Kandil, Passant, Sherin Moussa, and Nagwa Badr. "Regression testing approach for large-scale systems." Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on. IEEE, 2014.
2. Shahid, Muhammad, Suhaimi Ibrahim, and Mohd Naz'ri Mahrin. "Code Coverage Information to Support Regression Testing." The International Conference on Informatics and Applications (ICIA2012). The Society of Digital Information and Wireless Communication, 2012.
3. Carlson, Ryan, Hyunsook Do, and Anne Denton. "A clustering approach to improving test case prioritization: An industrial case study." Software Maintenance (ICSM), 2011 27th IEEE International Conference on. IEEE, 2011.
4. Yoo, Shin, et al. "Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge." Proceedings of the eighteenth international symposium on Software testing and analysis. ACM, 2009.
5. Elbaum, Sebastian, Alexey G. Malishevsky, and Gregg Rothermel. "Test case prioritization: A family of empirical studies." IEEE transactions on software engineering 28.2 (2002): 159-182.
6. Bharati, Chandana, and Shradha Verma. "Analysis of Different Regression Testing Approaches." Analysis 2.5 (2013).
7. Kapfhammer, Gregory M. "Empirically evaluating regression testing techniques: Challenges, solutions, and a potential way forward." Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on. IEEE, 2011.
8. Yoo, Shin, and Mark Harman. "Regression testing minimization, selection and prioritization: a survey." Software Testing, Verification and Reliability 22.2 (2012): 67-120.
9. Blondeau, Vincent, et al. "Test case selection in industry: an analysis of issues related to static approaches." Software Quality Journal (2016): 1-35.
10. Pathania, Yamini, and Gurpreet Kaur. "Role of Test Case Prioritization based on Regression Testing using Clustering." International Journal of Computer Applications 116.19 (2015).
11. Biswas, Swarnendu, et al. "Regression test selection techniques: A survey." Informatica 35.3 (2011).
12. Chittimalli, Pavan Kumar, and Mary Jean Harrold. "Recomputing coverage information to assist regression testing." IEEE Transactions on Software Engineering 35.4 (2009): 452-469.
13. Rothermel, Gregg, and Mary Jean Harrold. "Analyzing regression test selection techniques." IEEE Transactions on software engineering 22.8 (1996): 529-551.
14. Beena, R., and S. Sarala. "Code coverage based test case selection and prioritization." arXiv preprint arXiv:1312.2083 (2013).
15. Huang, Sheng, Jun Zhu, and Yuan Ni. "ORTS: a tool for optimized regression testing selection." Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications. ACM, 2009.}