

## Survey on Parallel Test Prioritization

### Background:

In regression testing, the cost of test execution can not be ignored. For example, in rapidly continuous integration (CI) process, the cost of test execution is large, as a large number of test cases are executed at high frequency. Therefore, how to reduce the cost of regression testing has attracted extensive attentions from both academia and industry. In the literature, academia researchers attempted to alleviate the cost problem of regression testing through various optimization techniques. For example, test case prioritization is used to schedule the order of test execution to accelerate fault detection. In industry, more computing resources are used to improve the efficiency of test execution by distributing the workload of testing. The existing work on test prioritization is conducted in sequential scenario, and ignores some features which exists in parallel scenario (e.g., whether the computing resources are allocated reasonable). In this work, we propose some parallel test prioritization techniques by combining sequential test prioritization and parallel scenario. More specifically, we first conduct parallel test prioritization technique to get the optimization order of test execution that can detect faults faster. Then, all the tests are allocated to different computing resources based on its execution order. The purpose of this questionnaire is to investigate some questions about the application of parallel test prioritization. We hope that you can spend about 2-3 minutes to fill in this questionnaire! Thank you for your help!

Q1: How long have you been engaged in testing?

Options		Results
A	1 year	11
B	2-5 years	11
C	Over 5 years	10

Q2: In your work on testing, do you think the cost of regression testing is large, and the testing time should be reduced?

Options		Results
A	Yes	30
B	No	2

Q3: In your work on testing, if you use more computing resources to improve testing efficiency in parallel, do you think the execution time of tests over different threads should be close?

Options		Results
A	Consider	21
B	May consider	9
C	Do not consider	2

Q4: Based on the evaluation results on 54 open-source subjects, parallel test prioritization technique can reduce the time detecting the first fault by 20%, and reduce the average time

detecting all faults by 76%. If the proposed technique can achieve similar effectiveness in practice, do you consider to use our approach?

Options		Results
A	Consider	30
B	May consider	2
C	Do not consider	0

Q5: Note that parallel test prioritization technique needs to spend some time running the algorithm in order to achieve the optimal execution order of tests (e.g., the time cost of parallel test prioritization is 48 seconds on average), the proposed technique can be conducted offline when you are not working. Do you consider to use our approach?

Options		Results
A	Consider	28
B	May consider	4
C	Do not consider	0

Q6: In your work, does the code evolve significantly between different code commits?

Options		Results
A	Large	5
B	Not large	27
C	Small	0
D	None	0

Q7: Parallel test prioritization technique generates the optimal order of test execution based on the code coverage collected from a historical revision, and the tests are executed according to this order every-time the code is committed. Therefore, the optimal effectiveness (fault detection) can not be achieved in regression testing. Many existing work has shown that test prioritization based on historical code coverage can still work in terms of fault detection when the code is modified. Note that the fault detection of parallel test prioritization may not be the optimal due to historical code coverage, do you still consider to use our approach?

Options		Results
A	Consider	15
B	May consider	16
C	Do not consider	1

Q8: Do you have any suggestions on the effectiveness or the efficiency of parallel test prioritization?