Parallel Test Case Prioritization for Distributed System Using Search Algorithms

Team 6
Seyoung Song, Subeom Park, Yoonho Nam, Azret Kenzhaliev

Abstract

TODO

1. Introduction

- Regression Test Case Prioritization
- Parallel Test Prioritization
- Parallel Test Prioritization, but different CPU

2. Parallel Test Prioritization

- Problem Description
- Problem Definition
- Effectiveness Measure

3. Algorithms

3.1. Greedy Algorithms

TODO

3.2. Simulated Annealing

TODO

3.3. Genetic Algorithms

TODO

4. Empirical Study

4.1. Research Questions

- RQ1: Which algorithm is most effective in solving the parallel test prioritization problem?

- RQ2: How do the number of computing resources and the relative performance between them influence the performance of the parallel test prioritization techniques?

4.2. Experimental Design

1. Sequential Test Prioritization
    - $c = \{1\}$
2. Parallel Test Prioritization
    - $c = \{2, 4, 8, 16\}$
3. Asymmetric Test Prioritization
    - $1 : 2$
    - $1 : 3$
    - $1 : 4$
    - $1 : 1 : 1 : 1 : 4 : 4 : 4 : 4$

Table 1: An example of computing scenarios

Computing Scenario	Relative Performances
Sequential Test Prioritization	[1]
Parallel Test Prioritization ($c = 2$)	[1, 1]
Parallel Test Prioritization ($c = 4$)	[1, 1, 1, 1]
Parallel Test Prioritization ($c = 8$)	[1, 1, 1, 1, 1, 1, 1, 1]
Parallel Test Prioritization ($c = 16$)	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Asymmetric Test Prioritization ($1:2$)	[1, 2]
Asymmetric Test Prioritization ($1:3$)	[1, 3]
Asymmetric Test Prioritization ($1:4$)	[1, 4]
Asymmetric Test Prioritization ($1:1:1:1:4:4:4:4$)	[1, 1, 1, 1, 4, 4, 4, 4]

4.3. Subjects

TODO: Seyoung

Table 2: Open-source subjects from GitHub

ID	Subjects	SLOC	#Test	Time (s)
0	commons-cli	9053	192	3.064
1	dictomaton	4318	53	14.067
2	disklrucache	1921	61	2.364
3	efflux	5633	40	0.581
4	exp4j	5699	311	11.350
5	gdx-artemis	3607	35	0.483
6	geojson-jackson	1569	60	1.284
7	gson-fire	3566	91	3.249
8	jactor	6984	60	11.628
9	jadventure	5276	74	2.311
10	jarchivelib	2256	33	0.217
11	java-faker	8541	571	34.154
12	java-uuid-generator	4321	46	0.937
13	javapoet	9874	346	15.323
14	jsonassert	3476	150	1.641
15	jumblr	2970	103	0.905
16	lastcalc	7271	34	13.672
17	low-gc-membuffers	13099	51	1.784
18	metrics	6493	76	43.964
19	mp3agic	10037	495	4.815
20	nv-websocket-client	8617	73	1.014
21	protoparser	5545	171	4.752
22	restfixture	8243	290	6.716
23	skype-java-api	9749	24	15.720
24	stateless4j	2728	88	2.146
25	stream-lib	8756	142	443.206
26	xembly	3030	63	6.834

4.4. Results and Analysis

TODO

5. Conclusion

- Discussion
    - Comparison With Sequential Test Prioritization
    - Practical Concerns
    - Generalizability
- Comments from Professor
    - How long should the entire test take for there to be real gains in prioritization?
    - The time gain from prioritization becomes smaller as the number of compute resources increases, so it may not be meaningful if you already have a lot of compute resources.

References

[1] Z. Li, M. Harman, and R. M. Hierons, “Search Algorithms for Regression Test Case Prioritization,”IIEEE Trans. Software Eng., vol. 33, no. 4, pp. 225–237, Apr. 2007, doi: 10.1109/TSE.2007.38.

[2] J. Chen et al., “Optimizing test prioritization via test distribution analysis,”in Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Lake Buena Vista FL USA, Oct. 2018, pp. 656–667. doi: 10.1145/3236024.3236053.

[3] Q. Luo, K. Moran, L. Zhang, and D. Poshyvanyk, “How Do Static and Dynamic Test Case Prioritization Techniques Perform on Modern Software Systems? An Extensive Study on GitHub Projects,”IIEEE Trans. Software Eng., vol. 45, no. 11, pp. 1054–1080, Nov. 2019, doi: 10.1109/TSE.2018.2822270.

[4] J. Zhou, J. Chen, and D. Hao, “Parallel Test Prioritization,”ACM Trans. Softw. Eng. Methodol., vol. 31, no. 1, Sep. 2021, doi: 10.1145/3471906.