

----- REVIEW 1 -----

PAPER: 29

TITLE: Automatic Localization of Linearizability Faults

AUTHORS: Yang Chen, Zhenya Zhang, Peng Wu and Yu Zhang

Overall evaluation: 1 (weak accept)

Reviewer's confidence: 3 ((high))

----- Overall evaluation -----

This paper develops a method for checking (i.e., testing) for linearizability bugs where the search focuses on so called "critical data race sequences (CDRS)" in the implementation. I found the paper generally well written (barring the typos and issues below) and think that the paper could be accepted. The result, as far as I understand, is sound, and the development well explained.

Pros:

- + It addresses the problem of testing linearizability bugs, and thus applies to real implementations.
- + It develops a tool based on Java Pathfinder.
- + The linearizability fault problem is reduced to the detection of a high-level data race.
- + The tool manages to find several known bugs in existing algorithms.

Cons:

- The method is incomplete: Programs that are fault-free (wrt linearizability) may also have a high-level data race. Thus, I would have expected some description of where false-positives could arise.
- Timing information for the tools should have been presented. Since this is an incomplete method, I would have expected the search to be scale to more than two threads. Is this possible?

Please make the following changes.

- Statement of Theorem 1. Please restate as: "If there is a linearizability fault, then there is a high-level data race." Then the contrapositive (as stated in the proof becomes more obvious.)

- Please be more precise about the statement "composed of the same operations" in Defns 1 and 3.
- Please do not refer to  $\sqsubset_S$  as a "happens-before" relation. This confuses matters (e.g., in relation to weak memory). It would be better to call this the real-time order, as is common in the literature.

I think the authors could do more to analyse algorithms that haven't been reported in the literature already. For the algorithms analysed, it would be good to see references to the original reports that describe the bugs found. E.g., for SNARK, please reference:

S. Doherty et al: DCAS is not a silver bullet for nonblocking algorithm design. SPAA 2004: 216–224

----- Detailed comments -----

Intro: "hot-spotted" ----> "hot" (twice)

Pg 3, Related work, par 2, l1: "devoted on" --> "devoted to"

Pg 4, last par, l2 and l3: Fig numbers missing.

Pg 6, Ex 1, par 2, l2: "anyone" --> "any"

Pg 8, par after Defn 3, l1: "Intuitively, a P" --> "Intuitively, a program P"

Pg 8, par after Defn 3, l4: "linearizable in a P" --> "linearizable in a program P" (also fix this elsewhere, eg. Pg 10, l2)

Sect 5.1, l2. "Surely the linearizability of .... Besides, we define a state, ..." I found these two sentences confusing. Please rewrite.

Example 6, l1: "take a loot" --> "take a look"

----- REVIEW 2 -----

PAPER: 29

TITLE: Automatic Localization of Linearizability Faults

AUTHORS: Yang Chen, Zhenya Zhang, Peng Wu and Yu Zhang

Overall evaluation: 2 (accept)

Reviewer's confidence: 2 ((medium))

----- Overall evaluation -----

Authors present a technique that can locate a linearizability fault in concurrent systems. The analytic framework is well designed, and in general the paper is well written and easy to understand. I've included a few minor corrections below.

My only issue is that I wanted to know a little bit more about the subjects of empirical evaluation, mostly to get the sense of scalability. Table 1 does not contain much details of the subject themselves, nor the number of threads used for each program. As such, it is difficult to see how the scalability trend will look like, when either the size of the program or the number of threads increases. If authors can include these details for the camera ready version, I believe the paper will be much stronger.

Minor Corrections

- Please do not write "In [5, 6, 23]": instead, either use the author names directly ("Choi et al. [5]..."), or simply refer to "existing work" as the collective noun.
- Do not write "Figs." in the main text: spell it out to "Figure". Also, in the last paragraph of Section 2, authors missed a reference to a figure ("labeled by # in Fig. .").

----- REVIEW 3 -----

PAPER: 29

TITLE: Automatic Localization of Linearizability Faults

AUTHORS: Yang Chen, Zhenya Zhang, Peng Wu and Yu Zhang

Overall evaluation: 0 (borderline paper)

Reviewer's confidence: 4 ((expert))

----- Overall evaluation -----

This paper presents an algorithm to locate linearizability bugs in concurrent objects. The algorithm is based on a new notion of critical data race sequences (CDRS), which is defined on a fine-grained trace model where each memory access (read/write/CAS) is recorded. Then the algorithm constructs a labeled tree model for program execution, and try to find CDRSes and some key patterns, which indicate linearizability bugs. A prototype tool FGVT is implemented based on this algorithm. The experiments over 6 concurrent objects show the effectiveness of the idea.

The paper tries to address an important and challenging problem, and the idea seems interesting. However, the presentation is really problematic, which makes the technical details inaccessible. The problems are not just due to poor English. There are many typos and sloppy formalization in the presentation of the key technical details.

Below are some detailed comments:

- 1) In Def. 5, are the 3rd and 4th bullets saying the same thing? What do you mean by “no temporal relation”? Is the next bullet (the 4th) trying to define “no temporal relation”?
- 2) Page 9, 2nd paragraph below Def. 5, the  $\subset$  should be  $\subset_{pre}$  instead?
- 3) In Def. 6, what is SE? When you say “A CDRS satisfies that ...”, do you actually mean “A CDRS SE satisfies that ...”? Otherwise SE comes from nowhere. In the two bullets in the definition, what is the trace following  $\subset_{pre}$  ? Is anything missing here? This is supposed to be one of the most important definitions in the paper. Why is it presented in such a sloppy way?
- 4) Example 4: what do you mean by “#1 wins”? According to you definition before, you only say “e1 wins e2 \*\*with respect to ... Sf \*\*”. Here which sf are you referring to?
- 5) Algorithm 1 is unreadable: what is supposed to be the input to the algorithm? The  $<$  relation cannot exist itself without mentioning a trace.

Which trace are you referring to? What is  $e'$ ? I see some natural language explanation of  $e'$  (which also referring to some trace), but you have to formalize it in your algorithm.

6) Fonts in Figs. 6–8 are too small, which make the figures unreadable.

7) I still don't see how you could locate the bug in the code. It seems you are detecting bugs. If this is true, then the title might be misleading.

8) There are too many grammatical errors which I cannot list here. Since some of the coauthors are supposed to have reasonable English background, please proofread the paper carefully.

----- REVIEW 4 -----

PAPER: 29

TITLE: Automatic Localization of Linearizability Faults

AUTHORS: Yang Chen, Zhenya Zhang, Peng Wu and Yu Zhang

Overall evaluation: 2 (accept)

Reviewer's confidence: 3 ((high))

----- Overall evaluation -----

This paper proposes a novel notion of critical data race sequence (CDRS) to analyze the concurrent program faults that cause violation of linearizability. CDRS is based on fine-grained trace model, which analyzes the memory states of an operation sequence. The authors present a very detailed definition of CDRS with concrete examples. They also develop a prototype tool to detect the real-world Java concurrent programs by CDRS. Authors claim that their localization technique is effective in analyzing the root causes of linearizability faults.

+ The paper addresses an important problem in program analysis.

+ The proposed method is reasonable in analyzing the causes of linearizability faults.

– The paper is hard to follow in some parts.

– Experiments are not convincing enough to reveal the effectiveness of the proposed method.

I find that this paper is hard to follow. I notice that the authors try to

make the proposed notion CDRS clear and present several examples along with the formal definition. However, there are at least 20 abbreviation and notations in this paper. I think it is better to give a symbol table to help readers follow the definitions.

In Motivation, authors show the drawbacks of previous studies to motivate their fine-grained trace model. I think before presenting the concrete definition of their method, the authors should present the intuitive idea of their method and explain why such intuitive idea would work.

There are some flaws in the experiments. Actually, I cannot understand how effective CDRS is after reading the experiment parts. The authors select six programs to evaluate their method. Why they select these programs? They should also present some basic information of the programs, such as the size, where is each test case from. Since some test cases are from prior work, it is better to compare this method to some previous studies. I note that there also exist other situations where the authors' method cannot point out exactly. What is the reason for that? I think the authors should explain more about the failure of their method and discuss what are the advantages and disadvantages in using this method in real scenarios.