

*论文题目: AtexRace: across thread and execution sampling for in-house race detection

*作者: 郭郁、蔡彦、扬子江

*单位: 美国西密歇根大学、中国科学院软件研究所

联系方式:

*文章发表信息: Proceedings of the 11th Joint Meeting of the European Software Engineering Conference and the ACM SigSoft Symposium on The Foundations of Software Engineering (ESEC/FSE), ACM, 2017 年 9 月, 315-325 页。

*原文链接地址: <https://doi.org/10.1145/3106237.3106242>

*正文:

1. 背景介绍

当两个或者多个线程同时对内存同一位置操作时, 其中, 至少有一个线程操作是写操作, 那么数据竞争就会发生。数据竞争是并程序的主要缺陷, 甚至在实际生活中引起巨大的问题。

```
Shared variables: int x = 0, y = 0; Lock m, n;
Input: <a, b>;

Thread t1
1. for (i=1 to 2 × a){
2.   if(i < a)   f1(i);
3.   else       f2(i);
4. }
5.
6. Function f1(i){
7.   acq(m)
8.   x += i;
9.   y += i;
10.  rel(m)
11. }
12. Function f2(i){
13.  acq(n)
14.  y += i;
15.  rel(n)
16. }

Thread t2
17. for (j=1 to 2 × b) {
18.  if(j < b)   f3(j);
19.  else       f4(j);
20. }
21.
22. Function f3(j){
23.  acq(m)
24.  x += j;
25.  y += j;
26.  rel(m)
27. }
28. Function f4(j){
29.  acq(n)
30.  y += j;
31.  rel(n)
32. }
```

图 1: 数据竞争

图 1 中行 9 和行 30 会发生数据竞争, 另外行 25 和行 14 之间也会发生。静态数据竞争的检测有很好的延展性, 但是会有误报, 动态数据竞争的检测和静态相比, 误报会少很多, 不过系统开销会大很多。Fasttrack 通过减少大量的内存操作, 它已经是当前动态检测付出代价很少的工具。即便如此, 连续监测多线程程序的内存操作, Fasttrack 还是有 400%到 800%的系统开销。

采样技术通过选择性监测内存操作, 可以很好的降低系统开销。当前采样算法主要又两类: 第一类是通过在用户集来检测, 这类采样必须系统开销代价极小, 而且这一类基于采样率和系统开销限制, 通常只能检测很小的一部分数据竞争。第二类则是主要针对内部测试, 在软件发布之前, 开发者通常会大量用不同的或者相同的测试用例来测试软件。LiteRace 就是第二类采样检测工具, 它假设未检测到的数据竞争一般存在于一些不常被执行到的函数(冷区)里面, 所以其通过减少监测经常被执行的函数(热区)来降低系统开销。

2. 我们的工作

我们提出了一种新的采样方案叫做 AtexRace，主要思想是如果一个内存地址已经被一个线程多次操作，与此同时有其它线程首次对此内存地址进行操作，那么 AtexRace 会同时监测这两个线程。而且，在检测的过程中 AtexRace 会保存采样信息，以提供之后多次重复测试使用。从而达到在大量测试过程中（例如有很多测试用例时），每次测试的开销会逐渐下降，当并不会漏报数据竞争。

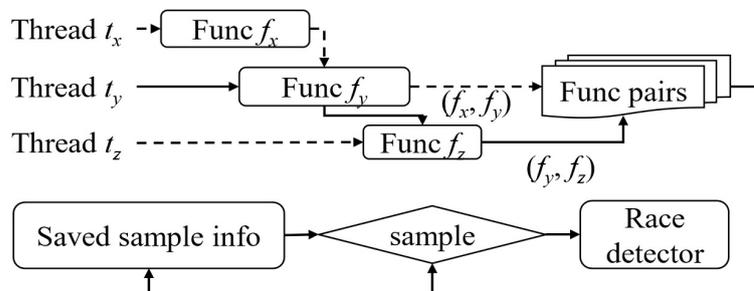


图 2: AtexRace 流程图

图 2 中当有函数对出现时，我们根据以后的采样信息来确定是否进行采样，并且同时更新采样信息。

3. 实验结果

我们分别在 Parsec 和 Mysql 上分别对 Fasttrack, LiteRace 和 AtexRace 做了对比试验：

Table 1. The statistics of Parsec benchmark and its overall results.

Benchmarks	Size (SLOC)	Pin	Time (seconds)			Overhead (%)			# of Unique Races		
			FT	LR	AR	FT	LR	AR	FT	LR	AR
Blackscholes	1,380	1048.53	1612.63	1652.39	1604.82	53.80%	57.59%	53.05%	0	0	0
Bodytrack	16,479	633.396	884.281	718.466	686.812	39.61%	13.43%	8.43%	14	14	32
Canneal	2,847	3314.22	7237.03	3640.18	4947.22	118.36%	9.84%	49.27%	2	2	2
Freqmine	2,192	2812.04	6451.47	4398.5	3317.82	129.42%	56.42%	17.99%	160	263	280
Streamcluster	1,795	111.626	135.767	131.325	131.915	21.63%	17.65%	18.18%	8	8	8
Avg.:						72.56%	30.98%	29.38%	Sum: 190	291	326

表 1 中，LiteRace 和 AtexRace 比 Fasttrack 快很多，此外由于采样算法会扰乱线程之间的调度，所以会出现 LiteRace 和 AtexRace 比 Fasttrack 检测到更多的去重数据竞争的现象。通过表一我们可以看出来，AtexRace 检测到了最多的数据竞争但是给系统带来的负载和 LiteRace 是几乎一样的。

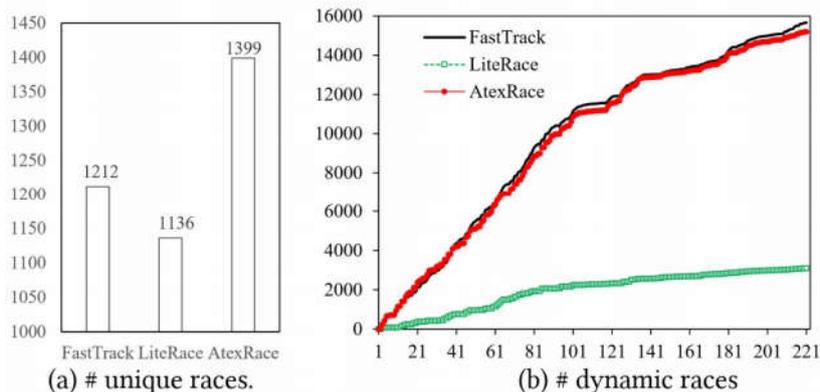


图 3: 数据竞争数目对比

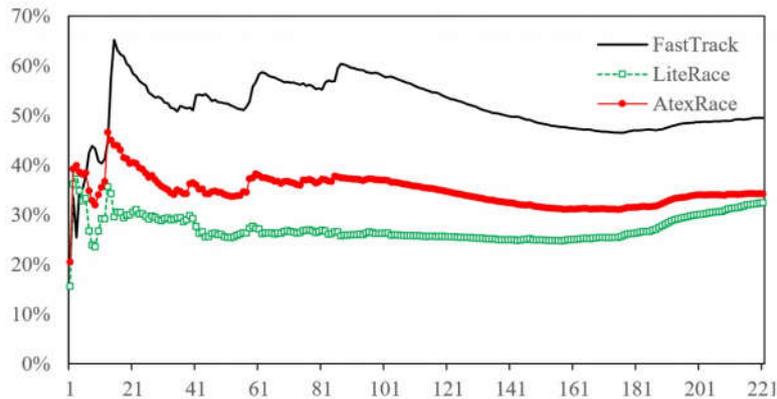


图 4: 系统负载对比

此外我们在 MySQL 上做的测试，图 3 和图 4 分别显示了三个工具对数据竞争检测的数目和系统负载的对比图，图三 (a) 中 AtexRace 依然检测到了最多的无重复的数据竞争，即使在数据竞争的总数量上也是和 Fasttrack 检测到的相差无几 (图三 b)。另外随着测试用例的增加，AtexRace 最终的系统负载已经几乎和 LiteRace 相同，我们相信如果有更多的测试用例的话，AtexRace 会最终超过 LiteRace。

4. 结论

我们实现了一个跨线程间和跨测试间的一个采样方法，同时还保证了数据竞争检测的高效性和高精度。通过实验确认 AtexRace 是一个很好的 Fasttrack 和 LiteRace 之间的一个替代品。

作者简介:

郭郁, 2013 年于西安交通大学大学获得学士学位, 当前是美国西密歇根大学大学的博士在读生。更多信息见 <http://sites.google.com/view/lei8845/>

蔡彦, 2014 年于香港城市大学获得博士学位, 之后在中国科学院软件研究所 (计算机科学国家重点实验室) 任副研究员。主要研究并发程序的测试问题, 并注重在大规模真实程序中的应用。近五年来发表文章 20 多篇 (CCFA 类 15 篇), 其中在 CCF A 类顶级期刊和会议 IEEE TSE、ICSE、FSE 上以第一作者身份发表 9 篇。是 COMPSACSETA 2018 Co-Program Chair, JSS(Journal of Systems and Software)的 EditorialBoard 成员, FCS (Frontiers of Computer Science) 期刊 YoungAssociateEditor (青年 AE), 多次为 IEEE TSE/TR/TC/ TSC、JSS、The Computer Journal 等著名期刊审稿、担任多个国际会议 PC 等。获 FCS 期刊 2016 年优秀青年 AE 奖。更多信息见 <http://lcs.ios.ac.cn/~yancai>。

杨子江, 2003 年于美国宾夕法尼亚大学获得博士学位, 之前在美国莱斯大学及中国科技大学获得硕士及学士学位, 现任西密西根大学计算机科学系正教授。目前主要从事软件测试及验证领域的研究。迄今为止发表了六十余篇论文(包括 CCF 推荐 A 类论文十余篇和 B 类论文二十余篇)及十项美国专利。获得过 ACM TODAES 最佳期刊论文奖, PADTAD 最佳会议论文奖, 西密西根大学工程及应用科学学院研究成就奖等。担任过密西根大学访问教授, NEC 公司科研顾问, 美国自然科学基金评委, 能源部企业扶植基金评委等职。更多信息见 <https://cs.wmich.edu/~zjiang/>。

说明: *为必填项。