

## MUVI: Automatically Inferring Multi-Var

摘要：

在多种软件bug之中，语义和并发错误是测的。

文章提出的方法，是通过代码分析来检测量相关交互过程中出现的问题：(1) *inconsistent updates*—相关的变量没有被一致地进行更新  
*multi-variable concurrency bugs*—相关联的操作在并发的程序中没有得到保护

MUVI (MULTi-Variable Inconsistency)自动检

# Variable Access Correlations and Detecting Related Sequential Consistency Bugs

MUVI提出的背景：

最难检

1. 动机：

则多变

consistent

更新 (2)

同存取

测程序

软件的缺陷会很明显的降低软件的更新和存取问题，用variable access表示，意思是，相关的变量们是内在的，只有在同时被读取的时候才能的，他们在被更新时，也需要是同虽然这是一个很重要的问题，但是档中，这导致一个问题，一旦程序

semantic and

可依赖性。本文的研究主要在相关变量的 correlations 或者 variable correlations 来表的被关联在一起的，他们的存在不是孤立表达出一个完整的表述或者视图，相应时被更新，而不能脱离了各自单独更新。程序员却不会将 access correlation 写入到文交给了其他程序员，别的程序员无法意识

中多变量的相关性存取 (access) 关系, 现其中的不一致更新操作和其他concurrent测试环境包括Linux, Mozilla, MySQL和PostgreSQL。

通过静态程序分析和数据挖掘技术进行多相关性的推断

通过代码分析找出相关变量不一致更新的问题  
改进两个经典的竞争检测方法 (lock-set和happens-before), 用于检测多变量相关的竞争。

关于变量相关性的说明, 变量基本通过以

同时发  
ncy bug

多变量

的地方

数据

以下的

到这么个关系，从而很容易就会破  
些变量仅仅是语义上相关，而不是  
他们的相关性，从而无法检测这个  
这就会导致刚才提到的两个问题：  
一致地进行更新 (2) multi-variable  
发的程序中没有得到保护  
之前唯一一个解决multi-variable co  
法，一旦有相关变量在同一个lock  
将会被绑定到只能在同一个critical  
到作用，原因是片面的认为相关变  
同时，现有的很多concurrency bug检

- lock-set data race detectors :
- Simply doing race detection at a  
object进行检测. 但是无法处

坏这个关系，出问题。更糟的是，因为数据上的依赖，传统的编译器无法分析出问题。

(1) *inconsistent updates*—相关的变量没有被  
*concurrency bugs*—相关联的存取操作在并

*concurrency bugs*的办法是，基于锁解决方案  
*critical section*中被同时*access*了，那么他们  
*section*中被*access*，然而这个方法并不能起  
量一定得同时被*access*。

检测都没法对多相关变量进行检测：

只针对单一变量，无法检测多相关变量

*coarse granularity*：虽然可以针对共享的

理不同*object*中变量的相关，同时，也无法

处理同一object内局部变量的

- Atomicity violation bug detection

到对于多个变量的access, 但

一变量在考虑, 因而还是没

在所有相关变量的同步问题都正确

的最主要的原因就是, 本应在同一

的事务里处理了, 也就是没有保证

最近的工作, 例如AtomicSet 和Color

人工标识相关变量来解决这个问题

需要有能够自动化监测出相关变量

就是这个工具的设计实现的动机。

Inconsistent update bug :

相关。

n tools : 考虑分检查code region, 能够检查是在推断出code region的时候, 还是基于单去解决多相关变量问题。

的被处理了之后, 出现上述concurrency bug atom region的相关变量的操作被放到了不同原子性。

rama都注意到了这个问题, 然而却提出要, 这会带来很巨大的人工代价。因此, 很并且找到其中的concurrency bug的工具, 这



个是together的, 同时MaxDistance是可以调

Access Correlation :  $x$ 和 $y$ 有操作相关性,  
 $A1(x) \Rightarrow A2(y)$ , 当且仅当  $A1(x)$  和  $A2(y)$   
together at least MinSupport times and when  
 $A1(x)$  appears,  $A2(y)$  appears together with  
least MinConfidence probability, where MinS  
and MinConfidence are tunable parameters  
and  $A2$  can be, respectively, any of the three  
read, write or AnyAcc.

相关性推断的步骤:

(1) Access Information Collection

(2) Access Pattern Analysis

(3) Correlation generation, pruning and ranking

调整的

如果

appear

never

at

support

s, A1

re:

ing

For any  $\text{write}(x) \Rightarrow \text{AnyAcc}(y)$  cor-  
examine the violations to it. All the f  
only update  $x$  without accessing  $y$  are  
incon- sistent update bug candidates.

缺点: Of course, MUVI inconsistency  
detection cannot solve all the multi-va  
inconsistency problem. Since MUVI only  
access types (read or write) and not s  
variable values, both false positives and  
negatives could occur due to special va

Multi-Variable Race Detection :

由于经典的两种方法Lock-Set和Hap  
法只能检测单变量, 要实现对多变

relation, we  
functions that  
treated as

nt update bug  
variable  
y considers  
specific  
nd false  
variable values.

pens-before 万  
量的 data race

MUVI的结果：

以抽取的样本来说，VariableAccessCorrection的方法进行处理；（2）某些变量是恰好的，同样也会存在漏报的现象，有以下的原定的情况下才会成立。

Inconsistent Update Bug Detection：确实有例子，有些情况下，相关的变量确实不需要update bug detection所使用导致出错；（3）变量确实存在漏报现象，主要有两个原因：（1）有些变量因此被忽略了。。

的检测，文章决定在这两种方法的修改。

relation的误报率在17%左右，主要的原因是 (1) 好才同时被更新，而不是真正的存在着相关性  
原因：(1) 有些真正的相关性由于数量太少，

能够发现新的而且是真的bug，但是同时存在该  
需要同时更新；(2) 前一步的相关性分析出了  
确实违反了相关性进行更新，但是后续不存在双  
些真正的相关性没有被分析出来；(2) 一些真

基础上进行

代码中的macro和inline函数需要特殊

。

被忽略了；（2）有些相关性只有在特

误报的现象由于以下的原因：（1）特

问题，分析出了不存在的相关性，被

对其的读取操作，不会造成错误。

真正的bug在bug报告里面被排的很后，

## Concurrency Bug Detection :

误报主要有两个原因：(1) 分析出了

漏报的原因：有些相关性只有在特定的

性能：Our MUVI extension only adds a small overhead and 1%-21% for happens-before.

结果分析：access correlations do not exist  
the correlated variables should

总结----可以扩展的点：

- (1) Detect other types of multi-variable r
- (2) Improve MUVI correlation analysis and
- (3) Extend MUVI to analyze dynamic trac

不存在的相关性； (2) 无害的多变量竞争 (只  
同程序上下文才能得出，而MUVI无法检测

small percentage of extra overhead on the original r

t between any two random variables

ld be either used or updated consistently.

related bugs, such as read inconsistency, multi-varia

bug detec- tion accuracy via better code analysis.

es to get run-time correlation.



只有在特殊的情况下才会有竞争)

race detectors: 5.9%-40% for lock-set;

able atomicity violation bugs, etc.

(4) Evaluate more real-world applications.



