

MUVI: Automatically Inferring M

摘要：

在多种软件bug之中，语义和并发错误是
文章提出的方法，是通过代码分析来检测
行更新 (2) multi-variable concurrency bug
MUVI (MULTI-Variable Inconsistency) 自动检
concurrency bug

测试环境包括Linux, Mozilla, MySQL和Post.

MUVI提出的背景：

...

Multi-Variable Access Correlations and Detecting Re

最难检测的。

则多变量相关交互过程中出现的问题：(1) inco
s—相关联的存取操作在并发的程序中没有得到
测程序中多变量的相关性存取 (access) 关系，

greSQL。

Related Semantic and Concurrency Bugs

inconsistent updates—相关的变量没有被一致地进行保护

同时发现其中的不一致更新操作和其他

1. 动机：

软件的缺陷会很明显的降低软件的可依赖性。variable correlations 来表示，意思是，相关性。不能表达出一个完整的表述或者视图，相关性。虽然这是一个很重要的问题，但是程序员和编译器无法意识到这么个关系，从而在数据上的依赖，传统的编译器无法分析出来。这就会导致刚才提到的两个问题：(1) independent 关联的存取操作在并发的程序中没有得到保证。之前唯一一个解决 multi-variable concurrent access 了，那么他们将会被绑定到只能在一定得同时被 access。

同时，现有的很多 concurrency bug 检测都

- lock-set data race detectors : 只针对

续性。本文的研究主要在相关变量的更新和存
关的变量们是内在的被关联在一起的，他们的
应的，他们在被更新时，也需要是同时被更新，
员却不会将access correlation写入到文档中，这
很容易就会破坏这个关系，出问题。更糟的是，
他们的相关性，从而无法检测这个问题。

consistent updates—相关的变量没有被一致地进
列保护

ncy bugs的办法是，基于锁解决方法，一旦有
同一个critical section中被access，然而这个方法

没法对多相关变量进行检测：

单一变量，无法检测多相关变量

取问题，用variable access correlations 或者
存在不是孤立的，只有在同时被读取的时候才
，而不能脱离了各自单独更新。

导致一个问题，一旦程序交给了其他程序员，
，因为这些变量仅仅是语义上相关，而不是数

行更新 (2) multi-variable concurrency bugs—相

相关变量在同一个lock critical section中被同时
并不能起到作用，原因是片面的认为相关变量

、

別

、

、

、

、

、

、

、

、

、

、

、

、

、

、

、

、

- Simply doing race detection at a coarse

同时，也无法处理同一object内局部

- Atomicity violation bug detection tools

候，还是基于单一变量在考虑，因

在所有相关变量的同步问题都正确的被处

的操作被放到了不同的事务里处理了，也

最近的工作，例如AtomicSet 和Colorama者

人工代价。因此，很需要有能够自动化监

通过静态程序分析和数据挖掘技术进行多

通过代码分析找出相关变量不一致更新的

改进两个经典的竞争检测方法 (lock-set和

关于变量相关性的说明，变量基本通过以

Constant Correlation

granularity : 虽然可以针对共享的object进行相关变量的相关。

: 考虑分检查code region, 能够检查到对于多线程而还是没法解决多相关变量问题。

处理了之后, 出现上述concurrency bug的最主要也就是没有保证原子性。

注意到了这个问题, 然而却提出要人工标识来监测出相关变量并且找到其中的concurrency bug

多变量相关性的推断

的地方

(happens-before), 用于检测多变量相关的数字

以下的几种方式进行关联:

检测，但是无法处理不同object中变量的相关，
个变量的access，但是在推断出code region的时
的原因就是，本应在同一atom region的相关变
相关变量来解决这个问题，这会带来很巨大的
的工具，这就是这个工具的设计实现的动机。
据竞争。

量

- Constraint specification
- Different Representation
- Different Aspects
- Implementation-demand

两个变量之间的access correlations并不是指

Access Together : 使用源代码距离 (代码在
说, 如果在同一个函数的两个操作 (读或
MaxDistance是可以调整的

Access Correlation : x 和 y 有操作相关性,
whenever $A1(x)$ appears, $A2(y)$ appears tog-
parame- ters, $A1$ and $A2$ can be, respective
相关性推断的步骤:

简单的同时更新，而是根据实现的细节情况。

相隔的行数) 来衡量两个变量之间的access关系
(读或者写) 之间的源代码距离比MaxDistance小，并

如果 $A1(x) \Rightarrow A2(y)$, 当且仅当 $A1(x)$ 和 $A2(y)$
together with at least MinConfidence probability, where
MinConfidence is a constant, and Acc is any of the three: read, write or AnyAcc.

具体分配

人, 同时以函数作为基础uint观察。也就是
那么就认为他们两个是together的, 同时

appear together at least MinSupport times and
ere MinSupport and MinConfidence are tunable

(1) Access Information Collection

(2) Access Pattern Analysis

(3) Correlation generation, pruning and ranking

Inconsistent update bug :

For any $\text{write}(x) \Rightarrow \text{AnyAcc}(y)$ correlation
treated as inconsistent update bug candidate

缺点 : Of course, MUVI inconsistent update bug
access types (read or write) and not specific

Multi-Variable Race Detection :

由于经典的两种方法Lock-Set和Happens-before
基础上进行修改。

ing

, we examine the violations to it. All the function
tes.

ate bug detection cannot solve all the multi-variabl
variable values, both false positives and false neg

efore 方法只能检测单变量, 要实现对多变量的

s that only update x without accessing y are

e inconsistency problem. Since MUVI only considers
atives could occur due to special variable values.

data race的检测, 文章决定在这两种方法的基础

MUVI的结果：

以抽取的样本来说，VariableAccessCorrela
行处理；（2）某些变量是恰好才同时被
同样也会存在漏报的现象，有以下的原因
下才会成立。

Inconsistent Update Bug Detection：确实能
情况下，相关的变量确实不需要同时更新
出错；（3）变量确实违反了相关性进行
漏报现象，主要有两个原因：（1）有些
略了。。

Concurrency Bug Detection：

tion的误报率在17%左右，主要的原因是 (1)
更新，而不是真正的存在着相关性。

因： (1) 有些真正的相关性由于数量太少，被

够发现新的而且是真的bug，但是同时存在误报

析： (2) 前一步的相关性分析出了问题，分析

更新，但是后续不存在对其的读取操作，不会

真正的相关性没有被分析出来； (2) 一些真

代码中的macro和inline函数需要特殊的方法进

被忽略了； (2) 有些相关性只有在特定的情况

报的现象由于以下的原因： (1) 特例，有些
析出了不存在的相关性，被detection所使用导致
会造成错误。

正的bug在bug报告里面被排的很后，因此被忽

)

,

误报主要有两个原因：(1) 分析出了不
漏报的原因：有些相关性只有在特定的和

性能：Our MUVI extension only adds a sma
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 rel
- (2) Improve MUVI correlation analysis and b
- (3) Extend MUVI to analyze dynamic traces
- (4) Evaluate more real-world applications.

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

11 percentage of extra overhead on the original rac

between any two random variables

be either used or updated consistently.

ated bugs, such as read inconsistency, multi-variab

ug detec- tion accuracy via better code analysis.

to get run-time correlation.

有在特殊的情况下才会有竞争)

e detec- tors: 5.9%-40% for lock-set; and

le atomicity violation bugs, etc.

