



The Lamp and Aladdin: Extracting Variable Correlations and Detecting High-Level Data Races

Raluca Ada Popa, MIT
2006 CRA - Women in Computer Science Summer Internship

Advisor: Professor Yuanyuan Zhou, UIUC
Joint work with Shan Lu, UIUC

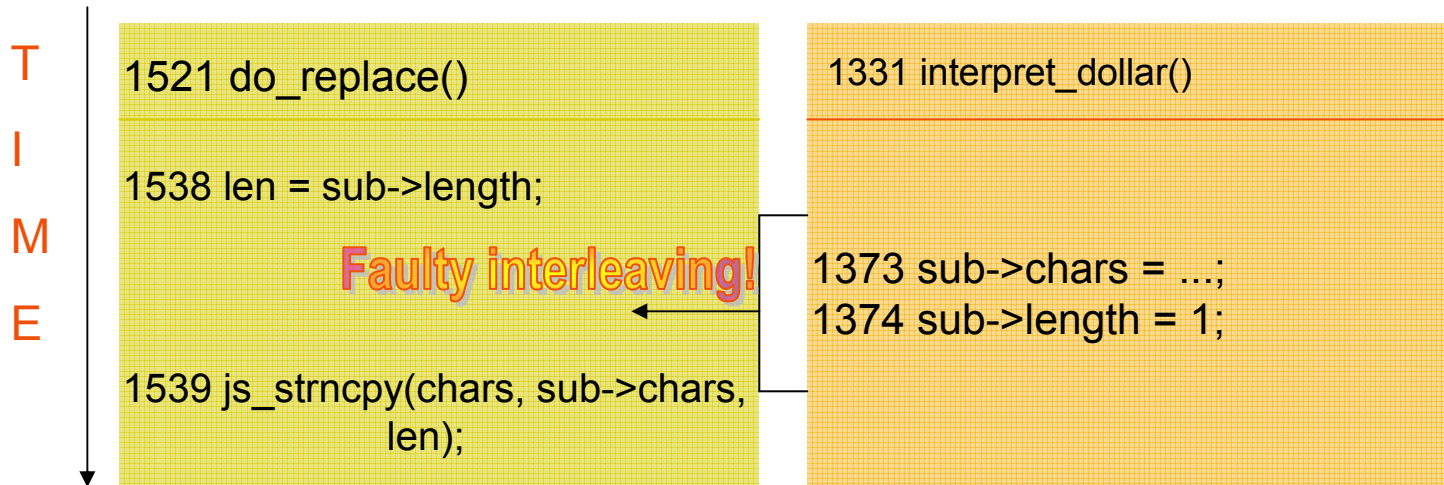
August 11, 2006

High-Level Data Races

- Races that disrupt the correlation between shared variables

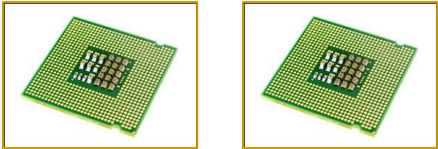
Mozilla Bug 142651 (simplified)

Correlation: the field **length** indicates the number of characters in the field **chars**



- Might not be a single shared variable data race
- This term is borrowed from Armin Biere & Co's paper

Motivation

- Multi-core technology is uprising 
- Transactional memory has gained many adepts
- Concurrent programming is prone to data races
 - Single variable races – AVIO, happens-before, lockset, SVD
 - High-level races – **open problem**

Our Contributions



The Lamp: The Variable - Correlation Generator



Aladdin: Existing tools for single-variable races

AVIO

Happens - before

Autolocker

Eraser

Our Contributions



The Lamp: The Variable - Correlation Generator



Aladdin: Existing tools for single-variable races

AVIO

Happens - before

Autolocker

Eraser



The Lamp and Aladdin: Tools for detecting, protecting and warning against high-level data races.

AVIOPlus

EraserPlus

Happens-BeforePlus

AutolockerPlus

Outline

- Introduction
 - High-Level Data Races
 - Motivation
 - Our Contributions
- The Variable Correlation Generator
 - PR-Miner
 - Feasibility Analysis and Observations
- The High-Level Data Race Detectors
- Related Work
- What Remains to Be Done
- Conclusions
- Acknowledgements



The Correlation Generator

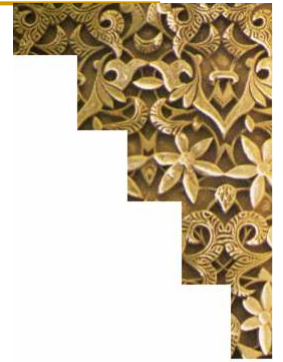


- Modify FPCLose implementation from PR-Miner
 - Mine correlations between shared variables
 - Rank by confidence and support

Confidence = (# functions **a** is with **b**, # functions **a** appears in)

- Is this enough?

Feasibility Analysis [1]



Bug 142651

Confidence of Chars

?

Confidence of Length

?

Feasibility Analysis[1]



Bug 142651

Confidence of Chars

31%

Confidence of Length

24%

- Obs.1: Add the structure type in front of fields before feeding to PR-Miner
 - Together: 4 writes and 1 read
 - Chars by itself: 11 reads
 - Length by itself: 16 reads
- Obs.2: It is highly likely for writes to occur together, while reads are less likely to be together.

Feasibility Analysis [2]



Bug 142651

Confidence of Chars

45%

Confidence of Length

36%

- Obs.1: Add the structure type in front of fields before feeding to PR-Miner
- Obs.2: It is highly likely for writes to occur together, while reads do not necessarily have to be together, so give lower weight to reads (1/2 in our calculations)

- In 11 functions:

```
343 chars = JSSTRING_CHARS(str);
```

```
344 length = JSSTRING_LENGTH(str);
```

Feasibility Analysis [2]



Bug 142651

Confidence of Chars 100%

Confidence of Length 80%



- Obs.1: Add the structure type in front of fields before feeding to PR-Miner
- Obs.2: It is highly likely for writes to occur together, while reads do not necessarily have to be together
- Obs.3: Inline macros and functions at least once

... and these hold for the other bugs we analyzed...

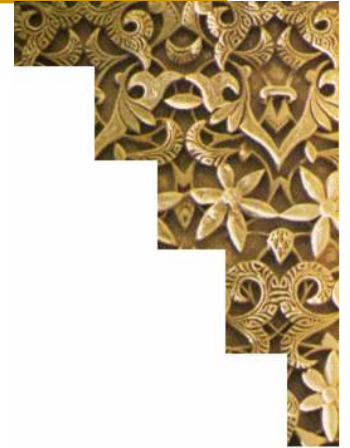
Outline

- Introduction
- The Correlation Generator
- The High-Level Data Race Detectors
 - AVIOPlus
 - EraserPlus
 - Happens-beforePlus
 - AutolockerPlus
- Related work
- What remains to be done
- Conclusions
- Acknowledgements





High-Level Data Race Detectors

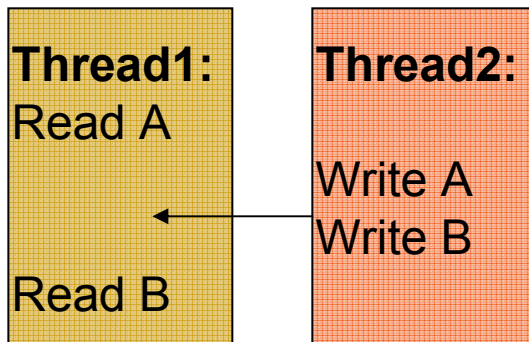


- Based on existing tools
- Use the variable correlations and some improvements:
 - AVIO \Rightarrow AVIOPlus
 - Eraser \Rightarrow EraserPlus
 - Happens-before \Rightarrow Happens-beforePlus
 - Autolocker \Rightarrow AutolockerPlus

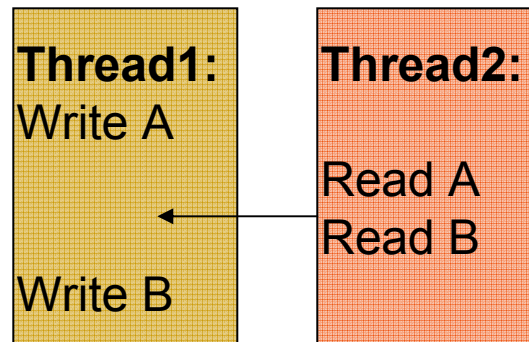
AVIOPlus

- AVIO detects single variable races
- We need new unserializable interleavings for each pair of correlated shared variables
 - There are 18 unserializable cases
 - The most common are:

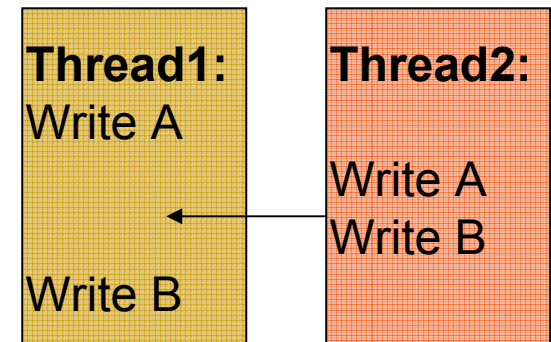
let A and B correlated



Mozilla: Bug 124651,
MySql: Bug 2011

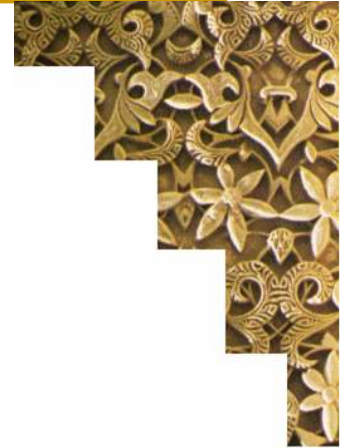


Mozilla: Bug 75305,
209188



MySql: Bug169, 13264,
14262, 14931

EraserPlus

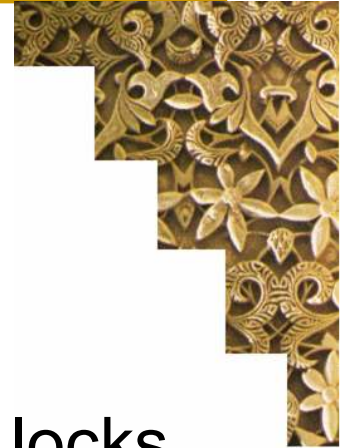


- Eraser warns when a shared variable is not consistently protected by a lock
- EraserPlus warns whenever two correlated variables are not consistently protected by a lock

Happens-beforePlus

- Happen-before tools claim data race when there is no happen-before order between accesses to the same variable
- Happen-beforePlus detects lack of happen-before order between accesses to shared variables

AutolockerPlus



- Autolocker implements atomic sections using locks
- AutolockerPlus can:
 - Warn when locks do not protect correlated variables
 - Ensure that correlated variables are protected by the same lock

Outline

- Introduction
- The Correlation Generator
- The High-Level Data Race Detectors
 - AVIOPlus
 - EraserPlus
 - Happens-beforePlus
 - AutolockerPlus
- Related Work
- What remains to be done
- Conclusions
- Acknowledgements



Related Work



- Focused on single variable data races: AVIO, SVD, happens-before, Eraser, Autolocker,...
- Few papers focused on multiple-variable data race:
“High-Level Data Races”
Cyrille Artho, Klaus Havelund, and Armin Biere
Computer Systems Institute, Switzerland
Kestrel Technology, NASA Ames Research Center,
PPoPP 2003
- Their tool detects 3 potential bugs, which seem to be false positives

What Remains to Be Done...

- Finish implementing the outlined improved tools
- Evaluate the tools
 - MySql, Mozilla
 - in comparison to the old tools

Contrast with related work, macar spune

Conclusions

Our work:

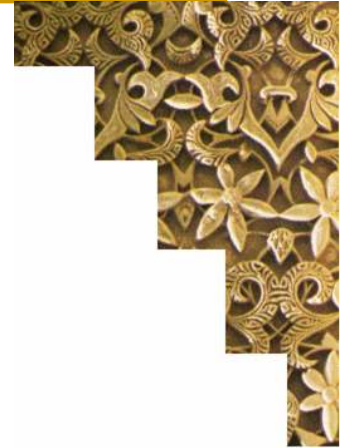
- Provides a variable correlation generator ... **a helpful tool**
- Improves existing work ... **used by the right objects**

➔ An effective way of determining multiple-variable data races

....**similar to the Lamp and Aladdin**



Acknowledgements



- Shan Lu
- Rebecca Shapiro
- Joe, Zhenmin, Pin, Qingbo, Feng, Xiao, Soyeon, Spiros, Hyojin

- Professor Liviu Iftode, Professor Yuanyuan Zhou

- Thank you all for the great summer!