Thread Safety for Third Party Libraries in Relaxed Memory Models

Ridhi Jain, IIIT-Delhi, India

Advisor:
Dr. Rahul Purandare, IIIT-Delhi, India

In Collaboration with:
Dr. Subodh Sharma, IIT-Delhi, India

Acknowledgements: Microsoft Research, FormaliSE
Motivation

Software crashes can be very expensive

- Failure of Mars Pathfinder
- Medical Machine Kills, Therac-25
- PayPal bug that made a $92 quadrillion deposit
Example

Data Race on variable x
Weak Memory Models

Weaker than Sequentially Consistent memory model

To enhance performance

**Total Store Order:** A store followed by load in a same thread on two different memory locations can be reordered

**Partial Store Order:** A store followed by load or a store in a same thread on two different memory locations can be reordered
Relaxed Memory

Thread 1
1. A = 1
2. print(B)

Thread 2
3. B = 1
4. print(A)

assert(A==1 || B==1)

Always true for sequentially consistent programs

May fail for relaxed memory models
Problem Statement

Let me use Libraries to save my software construction cost, time and effort!
Problem Statement

- These third party libraries can be faulty.
- The bugs can be even subtle in a multithreaded environment.
- May be even worse in Relaxed Memory models.
Example

PostgreSQL Database Server

Application
(Sharing connection object)

libpq

libpq
Motivation

*libpq*: Manipulation of same connection object, leads to a crash

*libcurl*: Has no internal thread synchronization

*libpng*: Using same instance of a structure might lead to a crash.
### Motivation

#### Figure 1: Thread safety status of sample libraries from Apache HTTP Server 2.x Thread Safety Issues

<table>
<thead>
<tr>
<th>Library</th>
<th>Version</th>
<th>Thread Safe?</th>
</tr>
</thead>
<tbody>
<tr>
<td>libcrypt</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Expat</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>FreeTDS</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>FreeType</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>GD 1.8.x</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>GD 2.0.x</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>gdbm</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>ImageMagick</td>
<td>5.2.2</td>
<td>Yes</td>
</tr>
<tr>
<td>libjpeg</td>
<td>v6b</td>
<td>?</td>
</tr>
<tr>
<td>OpenSSL</td>
<td>0.9.6g</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Research Question

Can we detect the harmful data races in a third party library in weaker memory models?
Questions that Follow

Are all the detected races harmful?
Can we detect data races on stripped binaries?
Can we detect the reorderable instructions in binaries?
Can we reorder stores and loads?
Related Work

- **A Promising Semantics for Relaxed-Memory Concurrency**
  

- **Dynamic Partial Order Reduction for Relaxed Memory Models**
  

- **Maple: A Coverage-Driven Testing tool for Multithreaded Programs**
  
System Overview

Binary File → Binary Race Detector for Relaxed Memory → Real Races

An Overview of the System
Detection of Shared Variables

Sample Program → Binary File → Variable Addresses → Local Variables and Shared Library Address Filter → Shared Variable Addresses
Detection of Data Races

Software Binary

Happens Before Relation and Store Buffers

Dynamic Binary Analysis

Racing Instructions

Shared Variable Addresses
Intermediate Results

- Tested on 16 libraries including boost, libcurl and libgpg.

- Detected data-races where **Intel Thread Checker** detected races, with additional reorderable instructions.

Not all these races are harmful!!
Real Races (Scheduler)

Dynamic Partial Order Reduction is used to replay the schedules.
Challenges

Binaries are to be analysed and dynamically controlled.

Debug information might not be available.

The implementation of application invoking library may be faulty.
Thank You