Automated Concurrency-Bug Fixing

Guoliang Jin, Wei Zhang, Dongdong Deng, Ben Liblit, Shan Lu

Concepts of Concurrent Computation seminar
Motivation

- Multi-core era
- Bug detectors are already proposed
- Fixing bugs is challenging
Bug Understanding

Fix-Strategy Design

Synchronization Enforcement

Patch Testing & Selection

Patch Merging
<table>
<thead>
<tr>
<th>Strategy (1)</th>
<th>Strategy (2)</th>
<th>Strategy (3)</th>
</tr>
</thead>
</table>

**Atomicity Violation**

Reports:

- **(a) All A → B**
- **(b) First A → B**

**Order Violation**

- **Bad**
- **Good**

**Race**

- **Remote-is-Bad**
- **Local-is-Bad**

**Def-Use**
Enforcing order relationship

- A instruction
- Signal thread
- S-create threads
- B instruction
- Wait thread
- W-create thread
- Call stack: \((f_0, i_0) \rightarrow (f_1, i_1) \rightarrow \cdots \rightarrow (f_n, i_n)\)
**allA-B order**

- locate places to insert signal operations in signal threads,

- locate places to insert signal operations in s-create threads,

- locate places to insert wait operations, and

- implement the signal and wait operations
Finding Signal Locations in Signal Threads

- Analyse Control Flow Graph (CFG) of $f_0$
- Insert signal operation on CFG edge that goes from reaching to non-reaching node
- Continue down the call stack
- Can $i_0$ call $f_1$ multiple times?
All A-B order

- locate places to insert signal operations in signal threads,
- locate places to insert signal operations in s-create threads,
- locate places to insert wait operations, and
- implement the signal and wait operations
Implementing Wait and Signal Operations

- Track the number of threads that will perform signal operation
- Track how many threads have signaled already
- Allow a wait thread to proceed once all signals are done
first A-B order

- Basic design
  - Signal right after A instruction
  - Wait right before B instruction

- Safety-net design
  - When program can no longer execute A, wait thread will continue
  - Inserts signal operations using all A-B algorithm
Bug Understanding

Fix-Strategy Design

Synchronization Enforcement

Patch Testing & Selection

Patch Merging
Correctness Testing

- Deadlock discovered by static analysis
- Failure in RTest
- Failure in GTest
- Timeout in RTest
- Failures of related patches
Patch selection

- Performance impact
- Number of synchronization operations
- Patch that can be merged
Patch merging guidelines

- The merged patch must have statically and dynamically fewer signal and wait operations than the unmerged patches.

- Each individual bug must still be fixed.

- Merging must not add new deadlocks.

- Merging should not cause significant performance loss.
## Overall results

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Bug Reports</th>
<th>Overall Patch Quality</th>
<th>Failure Rates</th>
<th>Overhead</th>
<th># of CFix Sync Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AV</td>
<td>OV</td>
<td>RA</td>
<td>DU</td>
<td>AV</td>
</tr>
<tr>
<td>OB1</td>
<td>2</td>
<td>5a</td>
<td>4</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>OB2</td>
<td>1</td>
<td>1a</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>OB3</td>
<td>7</td>
<td>4f</td>
<td>10</td>
<td>4L</td>
<td>✓</td>
</tr>
<tr>
<td>OB4</td>
<td>1</td>
<td>1f</td>
<td>2</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>OB5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>OB6</td>
<td>1</td>
<td>1f</td>
<td>2</td>
<td>1L</td>
<td>✓</td>
</tr>
<tr>
<td>OB7</td>
<td>1</td>
<td>1f</td>
<td>5</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AB1</td>
<td>6</td>
<td>6</td>
<td>1L</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AB2</td>
<td>1</td>
<td>2</td>
<td>1R</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AB3</td>
<td>2</td>
<td>4</td>
<td>2R</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AB4</td>
<td>1</td>
<td>2</td>
<td>3R</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AB5</td>
<td>4</td>
<td>5</td>
<td>1R</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>AB6</td>
<td>1</td>
<td>2</td>
<td>1R</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Limitations of CFix

- Only two different orderings
- Bugs including shared loops
- Some special scenarios...
Conclusion

- Different bug detectors
- Among first tools of this category
Thank you!

Questions?