Practical AJAX Race Detection
For JavaScript Web Applications

Christoffer Quist Adamsen, Anders Møller, Saba Alimadadi, Frank Tip
saba@northeastern.edu
JavaScript

<table>
<thead>
<tr>
<th></th>
<th>JS</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>popularity %</td>
<td>69.8</td>
<td>45.3</td>
</tr>
<tr>
<td>active repositories</td>
<td>323,938</td>
<td>222,852</td>
</tr>
</tbody>
</table>
function addRemoveFilter (fId) {
  filters = toggleFilter(fId);
  url = createUrl(server, filters);
  $.ajax({
    url: url,
    success: updateStationData
  });
}
function addRemoveFilter (fId) {
    filters = toggleFilter(fId);
    url = createUrl(server, filters);
    $.ajax({
        url: url,
        success: updateStationData
    });
}
function addRemoveFilter (fId) {
  filters = toggleFilter(fId);
  url = createUrl(server, filters);
  $.ajax({
    url: url,
    success: updateStationData
  });
}
Event Races
Event Races

Zheng et al., 2011
Petrov et al., 2012
Raychev et al., 2013
Hong et al., 2014
Jensen et al., 2015
Mutlu et al., 2015
Wang et al., 2016
Zhang & Wang, 2017
Adamsen et al., 2017a, 2017b
detecting harmful races
reducing benign warnings
providing helpful reports
performance and practicality
1. Generating event graphs

2. Testing potential conflicts
1. Generating event graphs

- Instrument automatically
- Trigger user events
- Generate event traces
- Create event graphs
- Find event conflicts

2. Testing potential conflicts
1. Generating event graphs

- Instrument automatically
- Trigger user events
- Generate event traces
- Create event graphs
- Find event conflicts

2. Testing potential conflicts
1. Generating event graphs

- Instrument automatically
- Trigger user events
- Generate event traces
- Create event graphs
- Find event conflicts

2. Testing potential conflicts

AXRacer
1. Generating event graphs

- Instrument automatically
- Trigger user events
- Generate event traces
- Create event graphs
- Find event conflicts

2. Testing potential conflicts
1. Generating event graphs

2. Testing potential conflicts

**synchronous mode**

- \( v \)
  - v’s consequences
  - \( w \)
    - w’s consequences

**adverse mode**

- \( v \)
  - w’s consequences
  - \( v \)
    - v’s consequences

potentially conflicted events
Does AJAXRacer help us find and understand AJAX races?
12 web applications, 20 pages
152 tests

- Pass: 80
- Fail: 65
- False Positive: 7

effectiveness
152 tests

effectiveness
false positives

- Pass: 80
- Fail: 65
- False Positive: 7

- Green: Pass
- Orange: Fail
- Red: False Positive
152 tests

- Pass: 80
- Fail: 65
- False Positive: 7

Effectiveness

Race characteristics

Dataset queries
Interactive maps
Auto completion
comparison

152 tests

- 80 Pass
- 65 Fail
- 7 False Positive

EventRacer

- 2 events
- ~130,000 races
- ~37,000 locations
- ~740 uncovered

Raychev et al. OOPSLA'13
https://eventracer.org/
performance

**phase 1**

<table>
<thead>
<tr>
<th>Duration (s)</th>
<th>Page Load</th>
<th>Execution</th>
<th>Avg: 26s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

**phase 2** (per test)

<table>
<thead>
<tr>
<th>Duration (s)</th>
<th>Page Load</th>
<th>Execution</th>
<th>Report</th>
<th>Avg: 69s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>36</td>
<td>28</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
common developer practices
The page contains diagrams and code snippets. The code snippet is: `function addRemoveFilter (fid) {
  filters = toggleFilter(fid);
  $.ajax({
    url: createUrl(server, filters),
    success: updateStationData
  });
}`

The diagrams illustrate the interaction between clients and servers, showing different states and transitions.