EventRacer: Finding Concurrency Errors in Event-Driven Applications

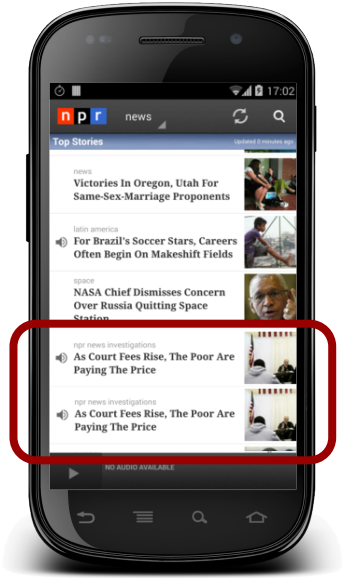
Pavol Bielik

ETH zürich

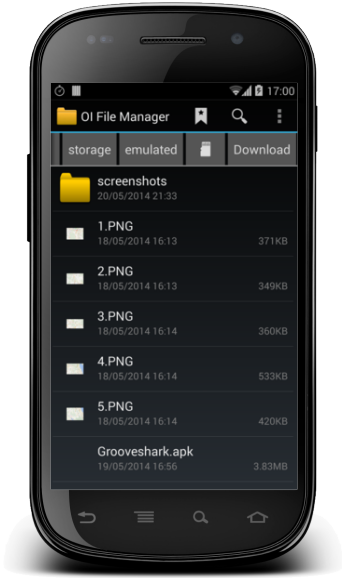
strangeloop
St. Louis, MO
Sept 17-19, 2014

SRL
SOFTWARE RELIABILITY LAB

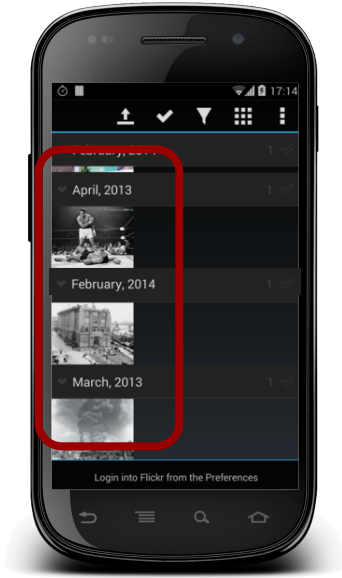
Android Errors Caused by Concurrency



Display article twice

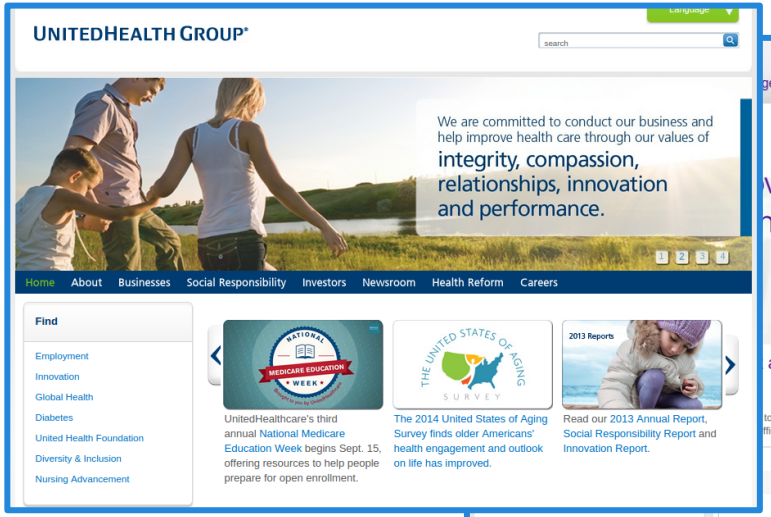


Display wrong directory

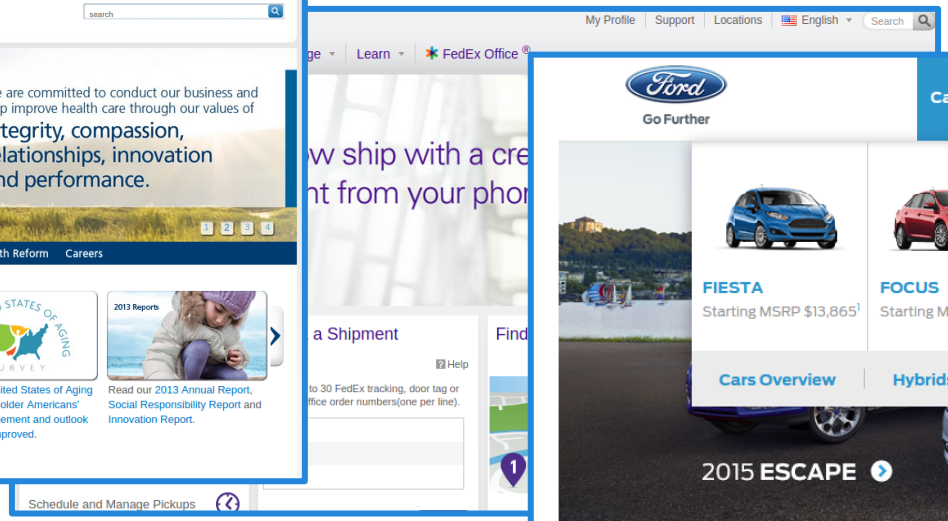


Display wrong order

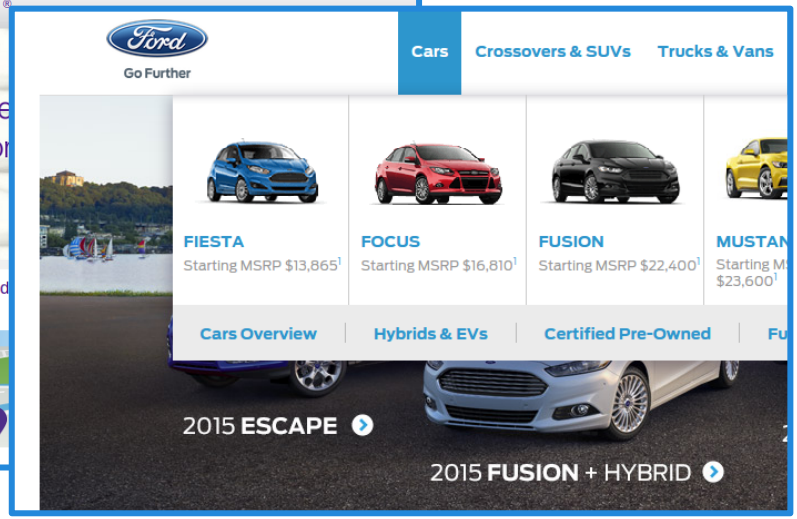
Web Page Errors Caused by Concurrency



Incomplete form submitted



jQuery version used non-deterministically



Non-operational menu

Event-Driven Applications

designed to hide latency, various asynchronous APIs

network, disk, database, timers, UI events

highly asynchronous and complex control flow

scheduling non-determinism

asynchrony is not intuitive

EVENT RACER

Trouble with Asynchrony

stackoverflow

- Background task, progress dialog, orientation change - is there any 100% working solution?
- JavaScript function sometimes called, sometimes not
- Avoiding race conditions in Google Analytics asynchronous tracking
- Ajax Call Sometimes Works, Sometime works and refreshes, Sometimes refreshes and fails …?
- Is AsyncTask really conceptually flawed or am I just missing something?

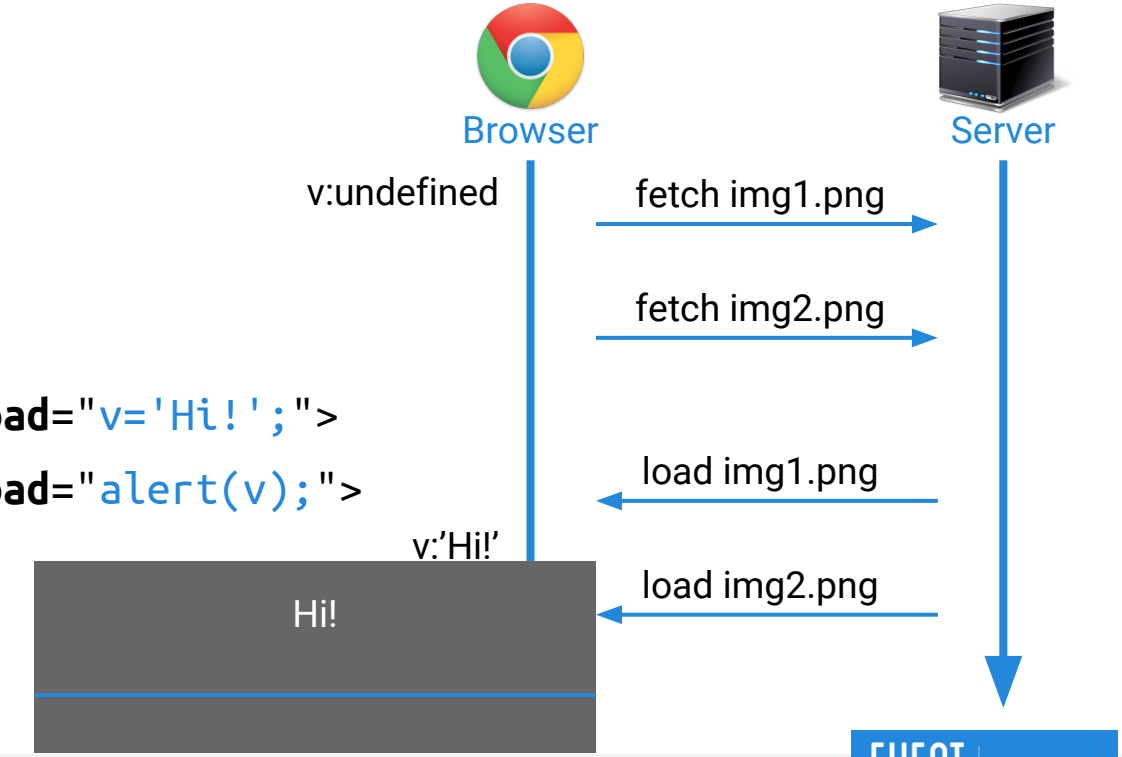
EVENT RACER

“Hello World” of web page concurrency

```
<html><body>
<script>
var v=undefined;
</script>

<img src="img1.png"    onload="v='Hi!';">
<img src="img2.png"    onload="alert(v);">


</body></html>
```

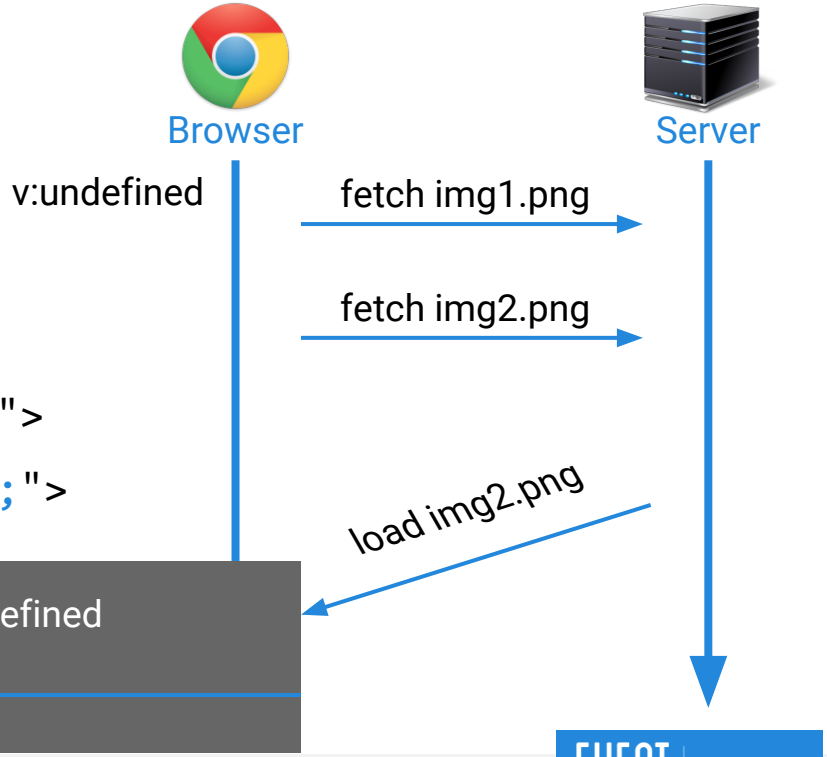


Bad interleaving

```
<html><body>
<script>
var v=undefined;
</script>

<img src="img1.png"    onload="v='Hi!';">
<img src="img2.png"    onload="alert(v);">


</body></html>
```



Understanding the problem

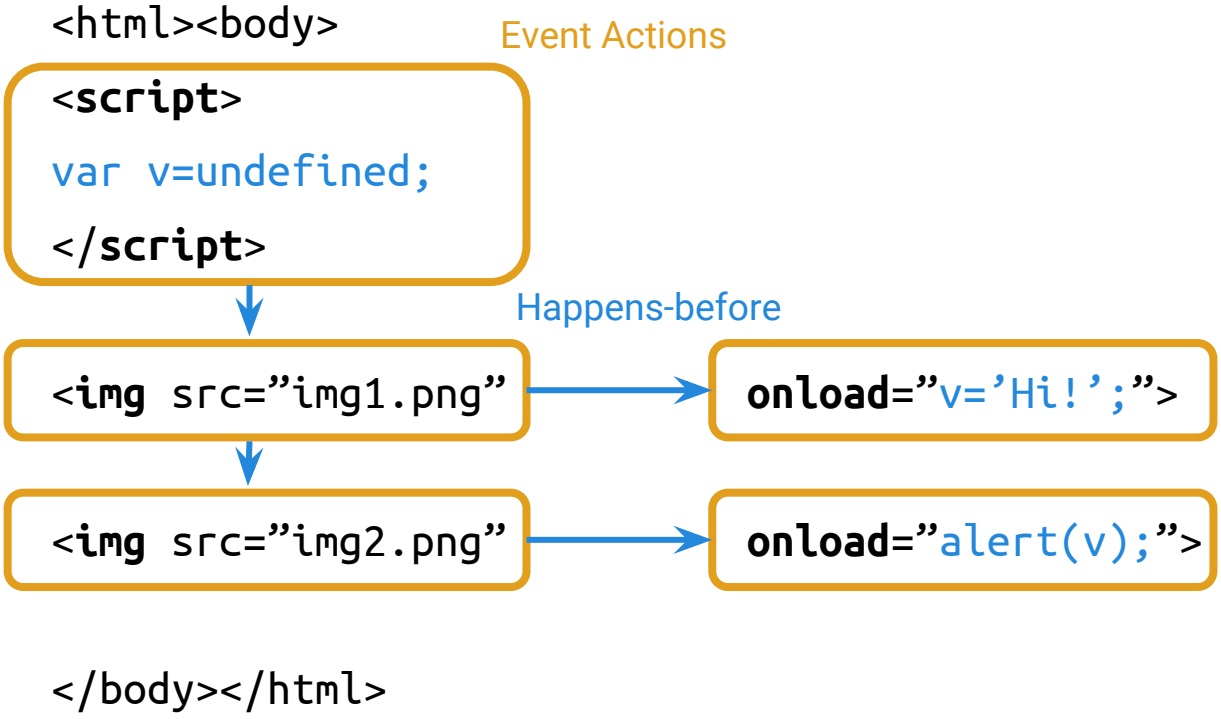
```
<html><body>
<script>
var v=undefined;
</script>
<img src="img1.png"
<img src="img2.png"
</body></html>
```

Event Actions

```
onload="v='Hi!';">
onload="alert(v);">
```

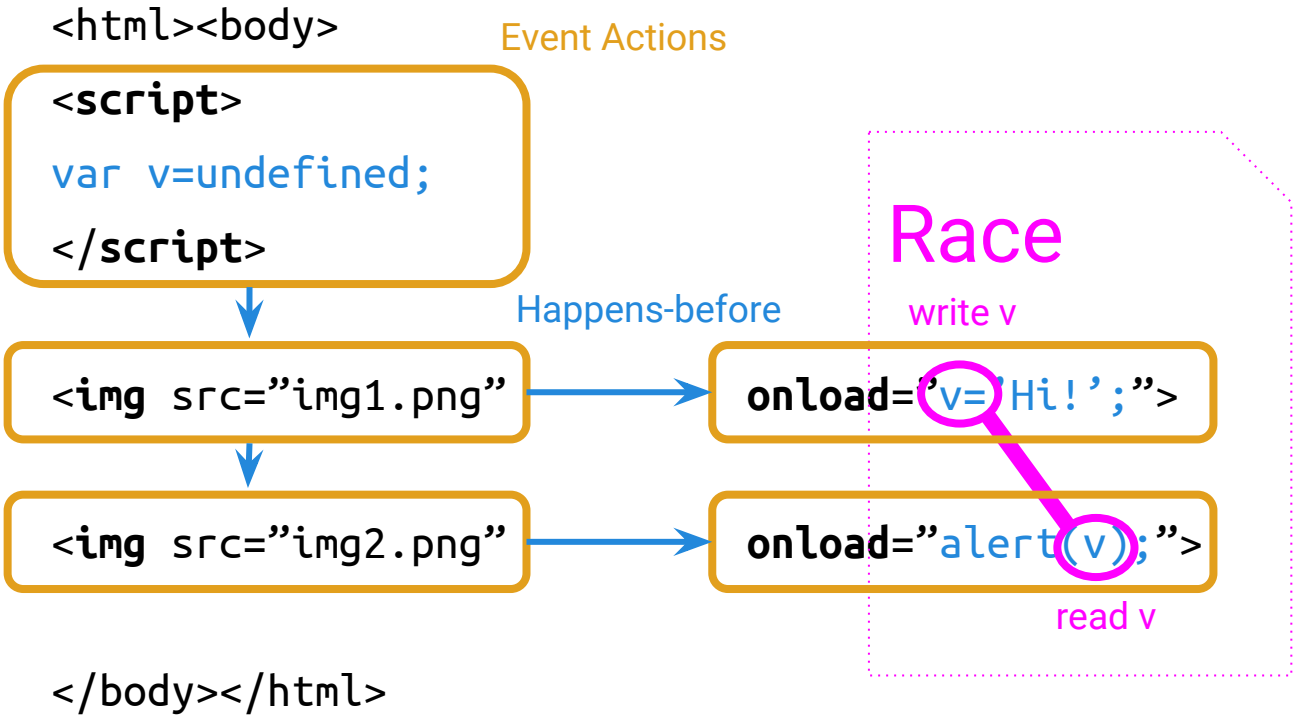

EVENT RACER

Understanding the problem



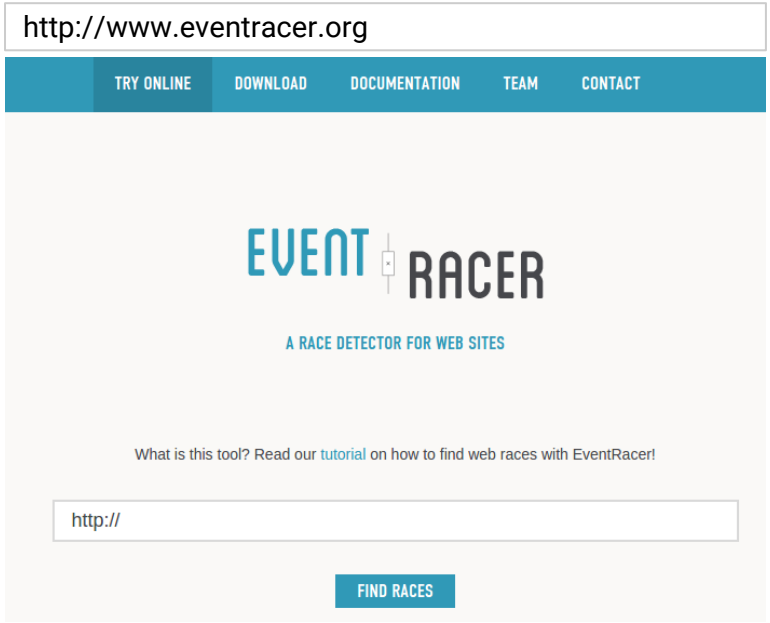
EVENT RACER

Understanding the problem



EVENT RACER

Online Analysis



Memory Locations

Search by name: Search

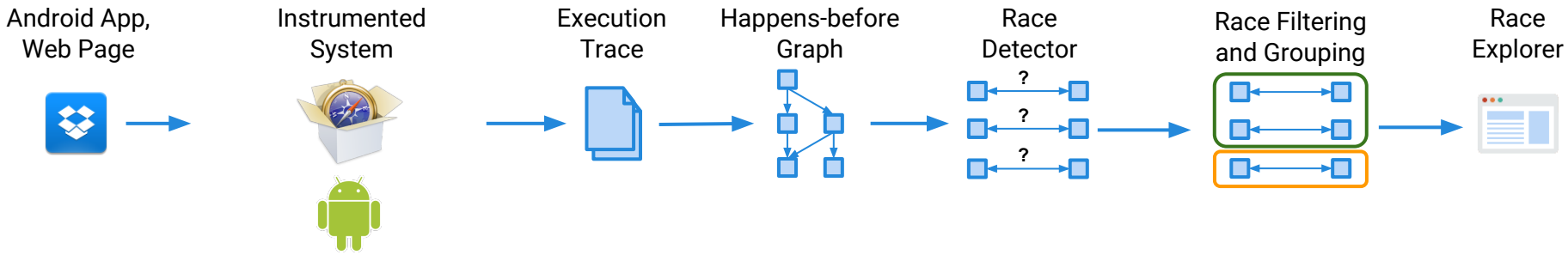
Shown memory locations: [only with races] **[only with uncovered races]** [only with uncovered unfiltered races] [only wi

Type	Name	Num. races	Num. uncovered races	Race classes
JS Variable	JSActivation[138074].c	2	1	readyStateChange race
JS Variable	Function[29].bind	3	1	
JS Variable	Window[26].gapi	3	1	
JS Variable	Window[26].___jsl	6	1	
JS Variable	[0x7f671ec27700].message	8	4	
JS Variable	Object[106535].ke	6	2	
JS Variable	Object[40173].plusone	4	1	
JS Variable	JSActivation[107403].Lp	1	1	
JS Variable	Object[54801]._isIframeResized	1	1	
JS Variable	[0x7f671ec27700].DOMContentLoaded	1	1	NO_EVENT_ATTACHED
JS Variable	#document[0x7f671ec36400].DOMContentLoaded	3	1	ONLY_LOCAL_WRITE
DOM Attribute	DOMNode[0x7f67180bece0].cookie	2	1	MAYBE_LAZY_INIT COOKIE_OR_CLASSNAME in
JS Array	Array[25294]$LEN	1	1	ONLY_LOCAL_WRITE
JS Variable	JSActivation[27873].g	7	1	MAYBE_LAZY_INIT initializa

14 rows

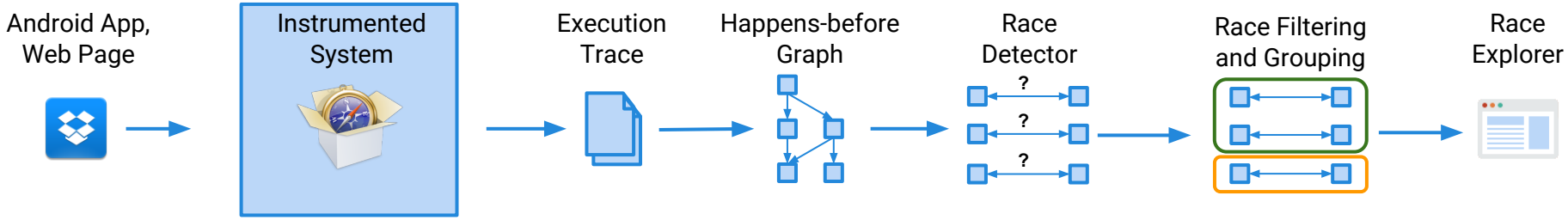
EVENT RACER

EventRacer end-to-end System



EVENT RACER

EventRacer end-to-end System



What are the memory locations on which asynchronous events can race?

JS variables, functions, arrays

```
<script>
v='Hi!';                    ↦ write(v)
function f() {}             ↦ write(f)
messages[2] = 42;           ↦ write(messages[2])
</script>
```

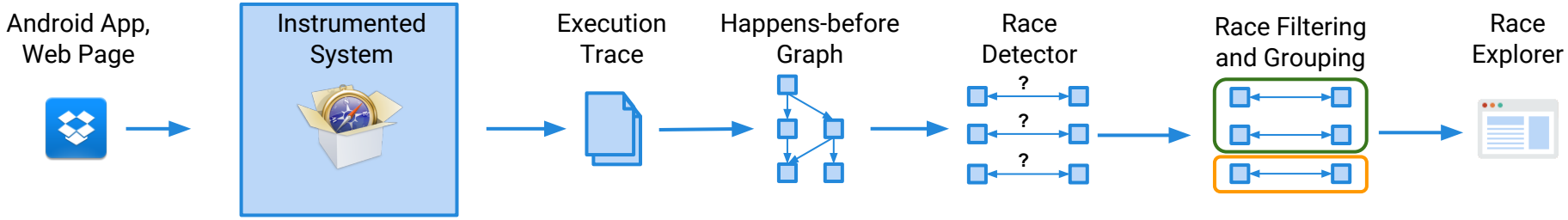
DOM nodes and attributes

```
<img id="img1" src="img1.png"        ↦ write(#img1)
     onload="v='Hi!';">              ↦ write(#img1.onload)

<script>
document.getElementById("img1")      ↦ read(#img1)
     .addEventListener("click", f);  ↦ write(#img1.click)
</script>
```


EVENT RACER

EventRacer end-to-end System



What are the atomic events used in event-driven applications?

Web
- parsing an HTML element
- executing a script
- handling user input
- ...

```
<script>
var v=undefined;
</script>
```

```
<img src="img1.png"
```

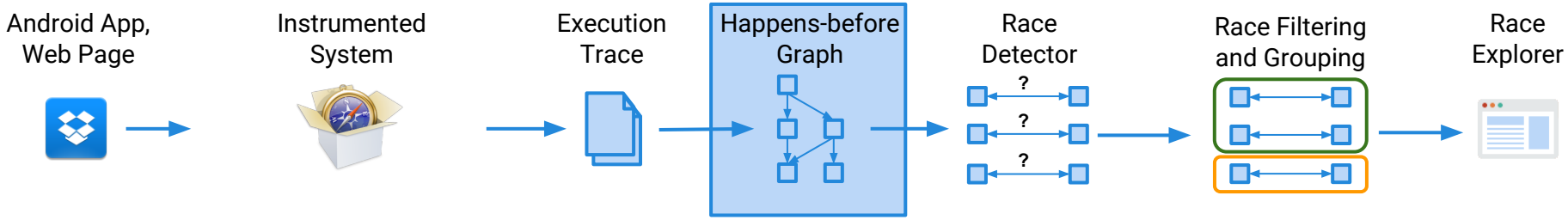
```
onload="v='Hi!';">
```

```
<img src="img2.png"
```

```
onload="alert(v);">
```


EVENT RACER

EventRacer end-to-end System



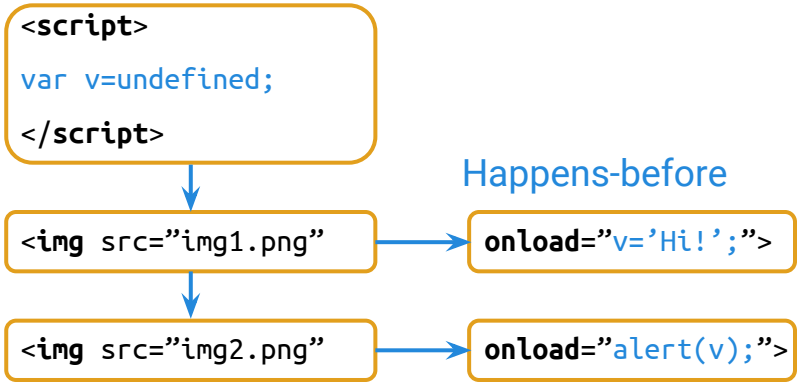
What is the event happens-before?

Web

```
setInterval, SetTimeout, AJAX, …
```

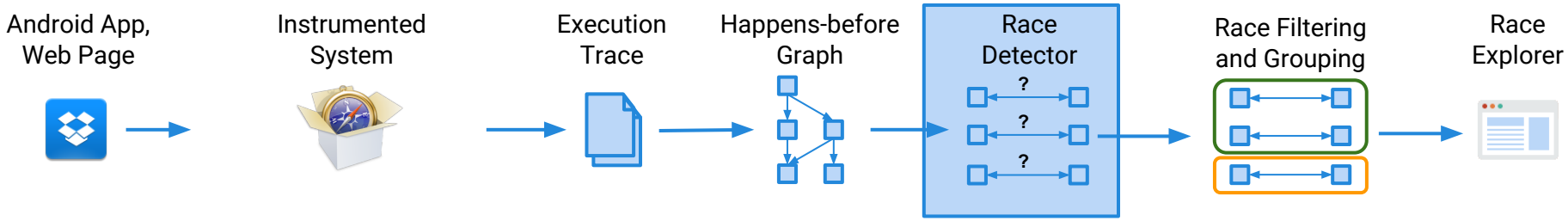
Android

```
postDelayed, postAtFront, postIdle, ...
```



EVENT RACER

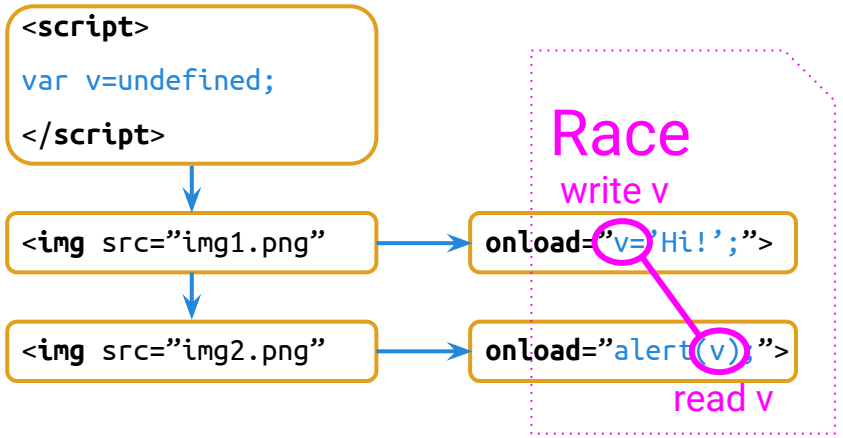
EventRacer end-to-end System



How to make scalable race detection in event-based setting?

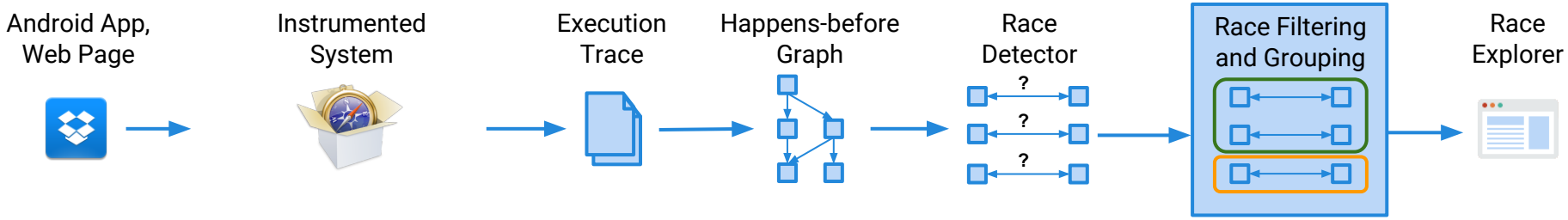
(Naive algorithms have asymptotic complexity $O(N^3)$ and require $O(N^2)$ space)

	State of the art	EventRacer
runtime	TIMEOUT	2.4sec
memory	25181MB	171MB



EVENT RACER

EventRacer end-to-end System



Is the system effective at finding harmful races while reporting few benign races?

We filter common classes of benign races:

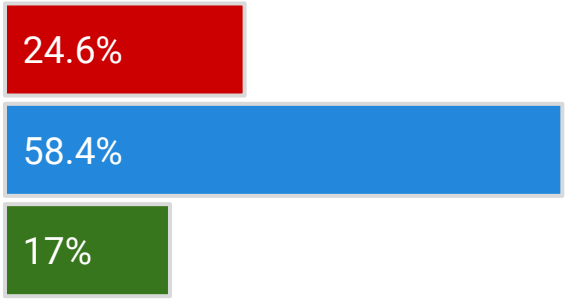
commutative operations, recycled objects, lazy initialization, local reads, ...

	Web	Android
# races found	646	1328
# races reported	17.3	13
reduction	**37x**	**100x**

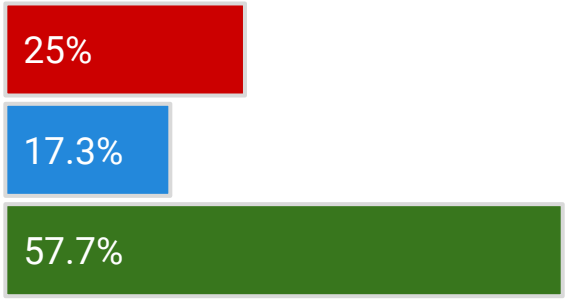
EVENT RACER

Manual evaluation

Web (314 reports)
Fortune 100 Web Pages



Android (104 reports)
8 Play Store Applications



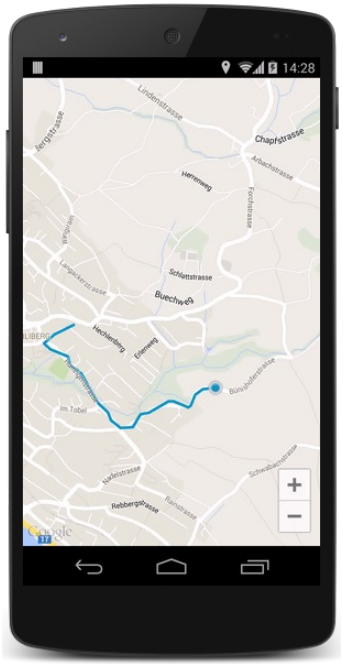
Harmful bugs
- ✓ unhandled exceptions
- ✓ UI glitches
- ✓ broken analytics
- ✓ page needs refresh to work normally

synchronization races
various idioms:
- ✓ if (ready) …
- ✓ try { … } catch { retry }
- ✓ array of callbacks
- ✓ etc.

harmless races
- ✓ commutative operations
- ✓ benign races
- ✓ framework related

EVENT RACER

Simple GPS Tracker

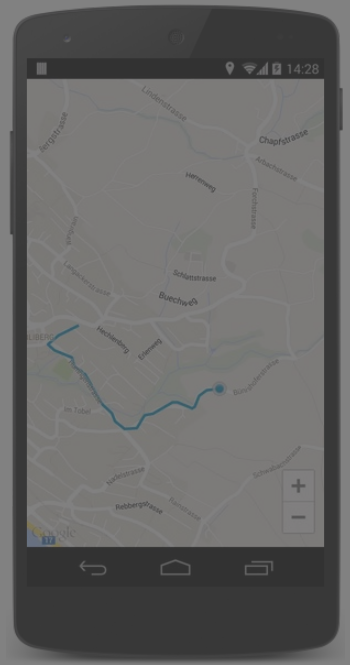



```
protected void onCreate() {
  locationManager.requestLocationUpdates(GPS_PROVIDER, 0, 0, mListener);
  mDbHelper = new SQLiteOpenHelper(this, DB_NAME, DB_VERSION);
}
```

```
LocationListener mListener = new LocationListener() {
  public void onLocationChanged(Location location) {
    //show location on map
    mDbHelper.getWritableDatabase().insert(loc);
  } };
```

```
protected void onStop() {
  locationManager.removeUpdates(mListener);
  mDbHelper.close();
}
```

Simple GPS Tracker



```
protected void onCreate() {
  locationManager.requestLocationUpdates(GPS_PROVIDER, 0, 0, mListener);
  mDbHelper = new SQLiteOpenHelper(this, DB_NAME, DB_VERSION);
}
```

```
LocationLis
  public vo
    //show
    mDbHelp
  } };
```

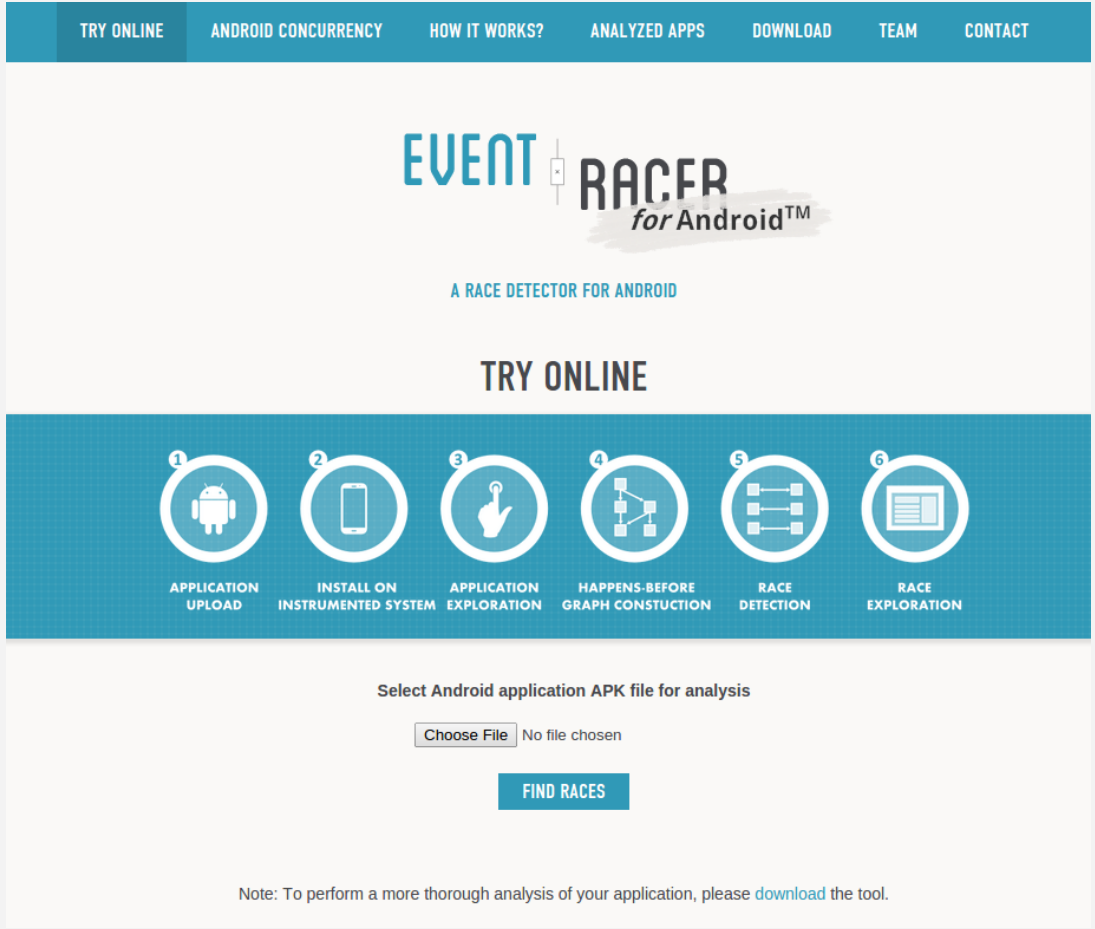
public void removeUpdates (LocationListener listener)

Added in API level 1

Removes all location updates for the specified LocationListener. Following this call, updates will no longer occur for this listener.

```
protected void onStop() {
  locationManager.removeUpdates(mListener);
  mDbHelper.close();
}
```

http://www.eventracer.org/android



EVENT RACER

Analysis Results

`onLocationChanged` and `onStop` are reported as not ordered

event 1 source	**event 2 source**
async IPC, interface(android.location.ILocationListener), code(onLocationChanged))	async IPC, interface(android.app.IApplicationThread), code (SCHEDULE_STOP_ACTIVITY))
→**calling context** Landroid/database/sqlite/SQLiteDatabase;.insert(...)	→**calling context** Landroid/database/sqlite/SQLiteClosable;.close(...)

→ UPDATE-UPDATE - 63568 Landroid/database/sqlite/SQLiteConnectionPool;.mAvailablePrimaryConnection
→ READ-UPDATE - 63568 Landroid/database/sqlite/SQLiteConnectionPool;.mIsOpen
→ READ-UPDATE - 63576 Landroid/database/sqlite/SQLiteConnection;.mConnectionPtr
→ READ-UPDATE - 63576 Landroid/database/sqlite/SQLiteConnection;.mPreparedStatementPool

EVENT RACER

Is the Alternative Interleaving Feasible?

```
D/GPS: onCreate
D/GPS: insert: Location[gps 47.284646,8.632389 acc=10 et=0 vel=2.0 mock]
D/GPS: insert: Location[gps 47.284656,8.632598 acc=10 et=0 vel=2.0 mock]
D/GPS: insert: Location[gps 47.284712,8.632722 acc=10 et=0 vel=2.0 mock]
D/GPS: insert: Location[gps 47.284832,8.632837 acc=10 et=0 vel=2.0 mock]
D/GPS: onStop
D/GPS: insert: Location[gps 47.285022,8.633205 acc=10 et=0 vel=2.0 mock]

E/AndroidRuntime: FATAL EXCEPTION: main
E/AndroidRuntime: Process: com.example.gps, PID: 2249
E/AndroidRuntime: java.lang.IllegalStateException: attempt to re-open an
already-closed object: SQLiteDatabase: /data/data/com.example.gps/test.db
```

Current Directions

Google Chromium port

- V8 javascript engine instrumentation

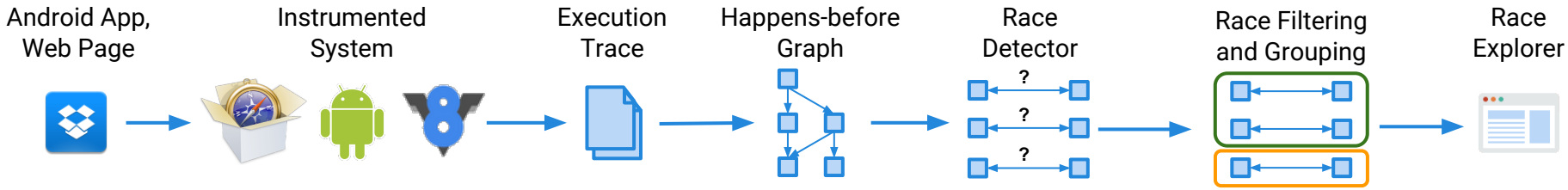
Testing tools based on EventRacer

- Integration with Selenium
- PhantomJS

Application for Parallelization

Other Application Domains (beyond Web Pages, Android)

- Node.js



www.eventracer.org

ETH zürich — Martin Vechev, Veselin Raychev, Pavol Bielik

AARHUS UNIVERSITY — Anders Møller, Casper Jensen

SAMSUNG RESEARCH AMERICA — Manu Sridharan

SU OF IA SOFIA UNIVERSITY — Boris Petrov, Yasen Trifonov

IBM WATSON — Julian Dolby

EVENT RACER