**Web Page Concurrency**

- **Event Actions**
  - `<script>
    var v=undefined;
  </script>`
  - `<img src="img1.png" onload="v='Hi!';">`
  - `<img src="img2.png" onload="alert(v);">`

**Instrumented System**

What are the memory locations on which operations can race?

- **JS VARIABLES, FUNCTIONS, ARRAYS**
  - `v='Hi!';` function `f() {}` messages[2] = 42;

- **DOM NODES, ATTRIBUTES**
  - `<img id="img1" src="img1.png" onload="v='Hi!';">` .addEventListener("click", f);

**Happens-before Graph Construction**

What is the event happens-before?

- **DOCUMENTATION**
  - `protected void onStop() { removeUpdates(mListener); mDbHelper.close(); }
  
protected void onCreate() {
    requestLocationUpdates(GPS, 0, 0, mListener);
    mDbHelper = new SQLiteOpenHelper(this, DB_NAME, DB_VERSION); 
    }

**Evaluation**

Is the system effective at finding harmful races?

Fortune 100 Web Pages (314 reports)

- HARMFUL: 24.6%
- SYNCHRONIZATION: 58.4%
- HARMLESS: 17.0%

8 Google Play Store Applications (104 reports)

- HARMFUL: 25.0%
- SYNCHRONIZATION: 17.3%
- HARMLESS: 57.7%

**Android Concurrency**

protected void onCreate() {
    requestLocationUpdates(GPS, 0, 0, mListener);
    mDbHelper = new SQLiteOpenHelper(this, DB_NAME, DB_VERSION); 
    }

**Race Detection**

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

- **VECTOR CLOCKS**
  - $O(1)$
  - $O(N^2)$
  - 544MB

- **VECTOR CLOCKS + CHAINED DECOMPOSITION**
  - $O(1)$
  - 0.04s
  - 5MB

**Race Filtering and Grouping**

How to produce feasible amount of reports?

- FILTERED
  - Write same value
  - Local reads
  - Lazy initialization

- REPORTED
  - Commutative operations
  - Race coverage
  - Recycled objects
  - Races inside framework

**Finding Bugs in Event-Driven Applications**

Pavol Bielik, ETH Zurich

Veselin Raychev, Pavol Bielik, Martin Vechev

Manu Sridharan

ETH Zürich

RESEARCH AMERICA