Thresher: Precise Refutations for Heap Reachability

Sam Blackshear    Bor-Yuh Evan Chang    Manu Sridharan
University of Colorado Boulder    IBM Research
Why does my phone crash when I rotate it?
Why does my phone crash when I rotate it?

Android: Crash on rotation, horizontal to vertical
Why does my phone crash when I rotate it?

Android: Crash on rotation, horizontal to vertical

Crash is detected after rotating phone in Gmail Sync now view a
1 post by 1 author 📈 3+1
Why does my phone crash when I rotate it?

Android: Crash on rotation, horizontal to vertical

Crash is detected after rotating phone in Gmail Sync now view
1 post by 1 author

phonegap
[important bug] cordova 1.9 crash on rotation android
5 posts by 2 authors
Why does my phone crash when I rotate it?

Android: Crash on rotation, horizontal to vertical

Crash is detected after rotating phone in Gmail Sync now view

phonegap

[important bug] cordova 1.9 crash on rotation android

5 posts by 2 authors

App crashes when rotating Samsung phone
Why does my phone crash when I rotate it?

Android: Crash on rotation, horizontal to vertical

Crash is detected after rotating phone in Gmail Sync now view a
1 post by 1 author

Phonegap
[important bug]cordova 1.9 crash on rotation android
5 posts by 2 authors

App crashes when rotating Samsung phone

Issue 20: Crashes when rotating phone horizontally
1 person starred this issue and may be notified of changes.
One source of rotation-based crashes:
Android Activity leaks
One source of rotation-based crashes: Android Activity leaks
One source of rotation-based crashes: Android Activity leaks

Android OS

Activity objects run the device’s UI
One source of rotation-based crashes: Android Activity leaks

Android OS

Activity-1

Activity objects run the device's UI
One source of rotation-based crashes:
Android Activity leaks

Activity objects run the device’s UI
One source of rotation-based crashes: Android Activity leaks
One source of rotation-based crashes: Android Activity leaks

Rotation triggers Activity destruction and re-creation
One source of rotation-based crashes: Android Activity leaks

Rotation triggers Activity destruction and re-creation
One source of rotation-based crashes: Android Activity leaks

Rotation triggers Activity destruction and re-creation
One source of rotation-based crashes: Android Activity leaks

- static_field
  - Object-1
    - fld
      - Activity-1
  - Activity-2
One source of rotation-based crashes: Android Activity leaks

```
static_field
  ↓
Object-1
  ↓
Activity-1
  ↓
Activity-2
```

Android OS
One source of rotation-based crashes: Android Activity leaks

- Held static reference to Activity-1 leads to memory leak.
One source of rotation-based crashes: Android Activity leaks

**static_field**

Object-1

fld

Activity-1

Can’t collect Activity-1!

Property to check:
No Activity is reachable from a static field.

Held static reference to Activity-1 leads to memory leak.
Detecting Activity leaks requires reasoning about heap reachability
Detecting Activity leaks requires reasoning about *heap reachability*

- Heap reachability: “Can an object be reached from another object or variable via pointer dereferences?”
Detecting Activity leaks requires reasoning about *heap reachability*

- Heap reachability: “Can an object be reached from another object or variable via pointer dereferences?”

- Need holistic view of the heap to query
Checking heap reachability with points-to analysis

Points-to analysis
Checking heap reachability with points-to analysis
Checking heap reachability with points-to analysis
Checking heap reachability with points-to analysis

Program → Points-to analysis → Fact database → Alarms
Checking heap reachability with points-to analysis
Checking heap reachability with points-to analysis

Program → Points-to analysis → Fact database → Alarms

- Fixes
- Manual triage
Checking heap reachability with points-to analysis

- Program
  - Points-to analysis
    - Fact database
      - Alarms
        - Manual triage
          - Fixes
            - Program
Checking heap reachability with points-to analysis

Program → Points-to analysis → Fact database → Lots of Alarms

Points-to analysis → Manual triage

Manual triage → Fixes

Fixes
Checking heap reachability with points-to analysis

Program → Points-to analysis → Fact database → Lots of Alarms

Manual triage
Real programs require extreme precision (path-sensitivity, context-sensitivity, and strong updates) to handle common programming patterns such as the null object pattern (see paper).
Real programs require extreme precision (path-sensitivity, context-sensitivity, and strong updates) to handle common programming patterns such as the null object pattern (see paper).
Real programs require extreme precision (path-sensitivity, context-sensitivity, and strong updates) to handle common programming patterns such as the null object pattern (see paper).
Program

Add *-sensitivity on-demand with symbolic analysis for *automated* triaging

Real programs require extreme precision (path-sensitivity, context-sensitivity, and strong updates) to handle common programming patterns such as the *null object pattern* (see paper)
Add *-sensitivity on-demand with symbolic analysis for *automated triaging*

Program → Points-to analysis → Fact database → Lots of Alarms

Manual triage → Fewer alarms → Thresher triage
Add *-sensitivity on-demand with symbolic analysis for *automated triaging*

Program

Points-to analysis

Fact database

Lots of Alarms

Fixes

Manual triage

Fewer alarms

Thresher triage
Program → Points-to analysis → Fact database → Lots of alarms

Fixes → Manual triage → Fewer alarms → Thresher triage
Advantages:
(1) Can be demand-driven w.r.t each alarm

Program

Fixes

Manual triage

Points-to analysis

Fact database

Lots of alarms

Thresher triage

Fewer alarms
Advantages:
(1) Can be demand-driven w.r.t each alarm
(2) Can utilize facts from up-front analysis to scale
Challenges

Why is adding \{path, context\}-sensitivity and strong updates on-demand hard?
Challenges

Why is adding \{\text{path, context}\}\text{-sensitivity and strong updates on-demand} hard?

- Control-path explosion
  - case splits for conditionals

\begin{align*}
Q_1 \lor Q_2 \\
\text{if } (\ldots) \{ \} \text{ else } \{ \}
\end{align*}
Challenges

Why is adding \{path, context\}-sensitivity and strong updates on-demand hard?

- Control-path explosion
  - case splits for conditionals

- Alias “path” explosion
  - aliasing case splits at every field write

\[ Q_1 \lor Q_2 \]
\[ \text{if (\ldots) \{\} else \{} \]
\[ x.f \]
\[ Q \]
Challenges

Why is adding \{path, context\}-sensitivity and strong updates on-demand hard?

- Control-path explosion
  - case splits for conditionals

- Alias “path” explosion
  - aliasing case splits at every field write

- Loops
  - need to infer path-specific loop invariants
Contributions

Fact database → Lots of alarms

Thresher triage

Monday, July 8, 13
Contributions

(1) Applying backwards automated triaging to refute false alarms by adding \{path, context\}-sensitivity and strong updates on-demand
Contributions

(1) Applying backwards automated triaging to refute false alarms by adding \{path, context\}-sensitivity and strong updates on-demand

(2) \textbf{from} constraints to leverage points-to facts, enabling scalability
Contributions

(1) Applying backwards automated triaging to refute false alarms by adding \{\text{path, context}\}-sensitivity and strong updates on-demand

(2) \textbf{from} constraints to leverage points-to facts, enabling scalability

(3) Per-path loop invariant inference over \textbf{from} constraints to soundly handle loops (see paper)
Contributions

(1) Applying backwards automated triaging to refute false alarms by adding \{path, context\}-sensitivity and strong updates on-demand

(2) \textbf{from} constraints to leverage points-to facts, enabling scalability

(3) Per-path loop invariant inference over \textbf{from} constraints to soundly handle loops (see paper)
Refute alarm by refuting edge

Leak Alarm

```
static_field
```

```
Object-1
```

```
fld
```

```
Activity-1
```
Refute alarm by refuting edge

Leak Alarm

static_field

Object-1

fld

Activity-1

Select an edge
Refute alarm by refuting edge

Leak Alarm

static_field

Object-1

fld

Activity-1

Select an edge

Perform backwards symbolic analysis to refute edge

Monday, July 8, 13
Refute alarm by refuting edge

Leak Alarm

static_field

Object-1

fld

Activity-1

Select an edge

Perform backwards symbolic analysis to refute edge

Refuted
Refute alarm by refuting edge

Select an edge

Perform backwards symbolic analysis to refute edge

Refuted

Eliminated false alarm!
Refute alarm by refuting edge

Leak Alarm

static_field

Object-1

fld

Activity-1

Select an edge

Perform backwards symbolic analysis to refute edge

Refuted

Not refuted

Eliminated false alarm!
Refute alarm by refuting edge

Select an edge

Perform backwards symbolic analysis to refute edge

Refuted

Not refuted

Eliminated false alarm!
Refute alarm by refuting edge

Leak Alarm

- static_field
- Object-1
- fld
- Activity-1

Select an edge

Repeat...

Perform backwards symbolic analysis to refute edge

Refuted

Eliminated false alarm!

Not refuted
Refute alarm by refuting edge

Fail to refute any edge in path = possible bug

Leak Alarm

static_field
Object-1
fld
Activity-1

Select an edge

Repeat...

Perform backwards symbolic analysis to refute edge

Refuted

Not refuted

Eliminated false alarm!
From points-to to program

static_field

Object-1

fld

Activity-1
From points-to to program

static_field

Object-1

fld

Activity-1
From points-to to program

Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

\[ Object_1 \cdot \text{fld} \rightarrow \text{Activity}_1 \]
From points-to to program

Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

Single instance of object

$Object_1.fld \rightarrow Activity_1$
Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”
Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

“What statement produces this query?”

Single instance of object

Exact points-to
Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

“What statement produces this query?”

\[ \text{obj.fld} = \text{act} \]

\[ \text{Object}_1 \cdot \text{fld} \rightarrow \text{Activity}_1 \]
Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

\[
\text{obj} \leftrightarrow \text{Object}_1 \ast \text{act} \leftrightarrow \text{Activity}_1
\]

\[
\text{obj.fld} = \text{act}
\]

\[
\text{Object}_1 \cdot \text{fld} \leftrightarrow \text{Activity}_1
\]
From points-to to program

Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

Strong updates via separation and exact points-to

\[
\text{obj} \mapsto \text{Object}_1 \star \text{act} \mapsto \text{Activity}_1
\]

\[
\text{obj.fld} = \text{act}
\]

\[
\text{Object}_1 \cdot \text{fld} \mapsto \text{Activity}_1
\]
From points-to to program

Pose query: “Some object instance allocated at Object-1 points to an object instance allocated at Activity-1 through fld”

Strong updates via separation and exact points-to

$\text{obj} \mapsto Object_1 \ast \text{act} \mapsto Activity_1$

$\text{obj.fld} = \text{act}$

$Object_1 \cdot \text{fld} \mapsto Activity_1$

Aliasing case splits!
if (a > b) {
    obj $\mapsto$ Object$_1$ $\ast$ act $\mapsto$ Activity$_1$
    obj.fld = act
    Object$_1$ $\cdot$ fld $\mapsto$ Activity$_1$
}
if (a > b) {
    \[
    \begin{align*}
    \text{obj} &\rightarrow \text{Object}_1 \times \text{act} \rightarrow \text{Activity}_1 \\
    \text{obj.fld} &= \text{act} \\
    \text{Object}_1 \cdot \text{fld} &\rightarrow \text{Activity}_1
    \end{align*}
    \]
From points-to to program

```java
obj → Object₁ * act → Activity₁ ∨ a > b
if (a > b) {
    obj → Object₁ * act → Activity₁
    obj.fld = act
    Object₁.fld → Activity₁
```
From points-to to program

Path-sensitivity by adding relevant path constraints

\[
\text{obj} \mapsto \text{Object}_1 \ast \text{act} \mapsto \text{Activity}_1 \land a > b
\]

if \((a > b)\) {

\[
\text{obj} \mapsto \text{Object}_1 \ast \text{act} \mapsto \text{Activity}_1
\]

\[
\text{obj.fld} = \text{act}
\]

\[
\text{Object}_1 \cdot \text{fld} \mapsto \text{Activity}_1
\]
From points-to to program

Path-sensitivity by adding relevant path constraints

Control-path explosion!

if (a > b) {
  obj ← Object₁ * act ← Activity₁ ∧ a > b
  obj.fld = act
  Object₁.fld ⟷ Activity₁
}
void foo(Object obj, Activity act) {
  if (a > b) {
    obj → Object₁ ∗ act → Activity₁ ∧ a > b
  }
  obj.fld = act
  Object₁ ∗ fld → Activity₁
}
From points-to to program

Context-sensitivity by forking
to all call sites

```
void foo(Object obj, Activity act) {
    obj ← Object_1 ∗ act ← Activity_1 ∧ a > b
    if (a > b) {
        obj ← Object_1 ∗ act ← Activity_1
        obj.fld = act
        Object_1 ∗ fld ← Activity_1
    }
}
```
void foo(Object obj, Activity act) {
    obj \rightarrow Object_1 \ast act \rightarrow Activity_1 \land a > b
}

if (a > b) {
    obj \rightarrow Object_1 \ast act \rightarrow Activity_1
    obj.fld = act
    Object_1 \cdot fld \rightarrow Activity_1
}

foo(y, z) \quad foo(w, x)
void foo(Object obj, Activity act) {
    if (a > b) {
        obj \rightarrow \textit{Object}_1 \ast \textit{act} \rightarrow \textit{Activity}_1 \land a > b
        obj.fld = act
    }
}

From points-to to program
Refute edge by refuting all paths

void foo(Object obj, Activity act) {
    if (a > b) {
        obj \rightarrow Object_1 \ast act \rightarrow Activity_1 \land a > b
        obj.fld = act
        Object_1 \cdot fld \rightarrow Activity_1
    }
}

From points-to to program
From points-to to program

Confirm edge by confirming one path

```plaintext
void foo(Object obj, Activity act) {
    obj \rightarrow Object_1 \ast act \rightarrow Activity_1 \land a > b
    if (a > b) {
        obj \rightarrow Object_1 \ast act \rightarrow Activity_1
        obj.fld = act
        Object_1 \cdot fld \rightarrow Activity_1
    }
}
```
Challenges

Control-path explosion!
- case splits for conditionals

Aliasing case splits!
- Need case splits at field writes to maintain separation
Challenges

Control-path explosion!
- case splits for conditionals

Can be tamed if # of relevant path constraints is small

Aliasing case splits!
- Need case splits at field writes to maintain separation
Challenges

Control-path explosion!
- case splits for conditionals

Can be tamed if # of relevant path constraints is small

Aliasing case splits!
- Need case splits at field writes to maintain separation
Challenges

Control-path explosion!
- case splits for conditionals

Can be tamed if # of relevant path constraints is small

Aliasing case splits!
- Need case splits at field writes to maintain separation

Handle by taking advantage of the up-front points-to analysis
Contributions

(1) Applying backwards automated triaging to refute false alarms by adding \{path, context\}-sensitivity and strong updates on demand

(2) \textbf{from} constraints to leverage points-to facts, enabling scalability

(3) Per-path loop invariant inference over \textbf{from} constraints to soundly handle loops (see paper)
\[ z \mapsto \widehat{Obj_1} \land (\widehat{Obj_1 \text{ from } r}) \]
Instance drawn from set of abstract locations $\mathcal{r}$

\[
z \mapsto \widehat{Obj}_1 \land (\widehat{Obj}_1 \text{ from } r)
\]
Example set:

\[
\hat{r} = \{ \alpha, \beta \}
\]

\[
\text{new}_\alpha \ Obj() \quad \text{new}_\beta \ Obj() \quad \text{new}_\gamma \ Obj()
\]

Instance drawn from set of abstract locations \( \hat{r} \):

\[
z \mapsto \widehat{Obj}_1 \land (\widehat{Obj}_1 \text{ from } \hat{r})
\]
Example set: \[ \hat{r} = \{ \alpha, \beta \} \]

\[ \text{new}_\alpha \text{Obj}() \]
\[ \text{new}_\beta \text{Obj}() \]
\[ \text{new}_\gamma \text{Obj}() \]

Instance drawn from set of abstract locations \( \hat{r} \)

\[ z \mapsto \hat{\text{Obj}}_1 \land (\hat{\text{Obj}}_1 \text{ from } \hat{r}) \]
Can refute if we derive \textit{false} via:

**Example set:**

\[ \mathcal{r} = \{ \alpha, \beta \} \]

\[ \text{new}_\alpha \text{Obj}() \]
\[ \text{new}_\beta \text{Obj}() \]
\[ \text{new}_\gamma \text{Obj}() \]

Instance drawn from set of abstract locations \( \mathcal{r} \)

\[ z \leftrightarrow \widehat{\text{Obj}_1} \land (\text{Obj}_1 \text{ from } \mathcal{r}) \]
Can refute if we derive \textbf{false} via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Example set:
\[ \mathcal{r} = \{ \alpha, \beta \} \]

Instance drawn from set of abstract locations \( \mathcal{r} \)

\[ z \mapsto \widehat{O bj_1} \land (\widehat{O bj_1 \ from \ \mathcal{r}}) \]
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. $\hat{a} > \hat{b} \land \hat{a} < \hat{b}$

Simultaneous points-to: e.g. $\hat{o} \cdot f \mapsto \hat{u} \ast \hat{o} \cdot f \mapsto \hat{v} \land \hat{u} \neq \hat{v}$

Example set:

\[ \mathcal{r} = \{ \alpha, \beta \} \]

Instance drawn from set of abstract locations $\hat{\mathcal{r}}$

\[ z \mapsto \hat{Obj}_1 \land (\hat{Obj}_1 \text{ from } \hat{\mathcal{r}}) \]
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Simultaneous points-to: e.g. \( \hat{o} \cdot f \leftrightarrow \hat{u} \land \hat{o} \cdot f \leftrightarrow \hat{v} \land \hat{u} \neq \hat{v} \)

From no allocation site: e.g. \( \hat{o} \text{ from } \emptyset \)

Example set:
\[ \hat{r} = \{ \alpha, \beta \} \]

\[ \text{new}_\alpha \text{ Obj}() \]
\[ \text{new}_\beta \text{ Obj}() \]
\[ \text{new}_\gamma \text{ Obj}() \]

Instance drawn from set of abstract locations \( \hat{r} \)

\[ z \mapsto \text{Obj}_1 \land (\text{Obj}_1 \text{ from } \hat{r}) \]
Can **refute** if we derive **false** via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Simultaneous points-to: e.g. \( \hat{\partial} \cdot f \mapsto \hat{u} \ast \hat{\partial} \cdot f \mapsto \hat{v} \land \hat{u} \neq \hat{v} \)

From no allocation site: e.g. \( \hat{\partial} \text{ from } \emptyset \)

\[ z = y \cdot f \]

\[ z \mapsto \widehat{Obj_1} \land (\widehat{Obj_1 \text{ from } \rho}) \]
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. $\hat{a} > \hat{b} \land \hat{a} < \hat{b}$

Simultaneous points-to: e.g. $\hat{\sigma} \cdot f \leftrightarrow \hat{u} \ast \hat{\sigma} \cdot f \leftrightarrow \hat{v} \land \hat{u} \neq \hat{v}$

From no allocation site: e.g. $\hat{\sigma}$ from $\emptyset$

$$z = y.f$$

$$z \leftrightarrow \widehat{Obj_1} \land (\widehat{Obj_1 \text{ from } r})$$
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Simultaneous points-to: e.g. \( \hat{\varnothing} \cdot f \mapsto \hat{u} \ast \hat{\varnothing} \cdot f \mapsto \hat{v} \land \hat{u} \neq \hat{v} \)

From no allocation site: e.g. \( \hat{\varnothing} \text{ from } \emptyset \)

Restricting **from** set based on precomputed points-to analysis

\[
(y \mapsto \widehat{Obj_2}) \ast (\widehat{Obj_2} \cdot f \mapsto \widehat{Obj_1}) \land (\widehat{Obj_1} \text{ from } \text{pt}(y.f) \cap \mathcal{R}) \land (\widehat{Obj_2} \text{ from } \text{pt}(y))
\]

\[
z = y \cdot f
\]

\[
z \mapsto \widehat{Obj_1} \land (\widehat{Obj_1} \text{ from } \mathcal{R})
\]
Can **refute** if we derive **false** via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Simultaneous points-to: e.g. \( \hat{\alpha} \mapsto \hat{u} \land \hat{\alpha} \mapsto \hat{v} \land \hat{u} \neq \hat{v} \)

**From no allocation site**: e.g. \( \hat{\alpha} \text{ from } \emptyset \)

**Restricting from** set based on precomputed points-to analysis

\[
X.f = p \\
(y \mapsto \text{Obj}_2 \mapsto (\text{Obj}_2.f \mapsto \text{Obj}_1) \land (\text{Obj}_1 \text{ from pt}(y.f) \cap \hat{r}) \land (\text{Obj}_2 \text{ from pt}(y))) \\
z = y.f \\
z \mapsto \text{Obj}_1 \land (\text{Obj}_1 \text{ from } \hat{r})
\]
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. $\hat{a} > \hat{b} \land \hat{a} < \hat{b}$

Simultaneous points-to: e.g. $\hat{o} \cdot f \mapsto \hat{u} \ast \hat{o} \cdot f \mapsto \hat{v} \land \hat{u} \neq \hat{v}$

From no allocation site: e.g. $\hat{o}$ from $\emptyset$

$x.f = p$

$(y \mapsto \textit{Obj}_2) \ast (\textit{Obj}_2.f \mapsto \textit{Obj}_1) \land (\textit{Obj}_1 \text{ from pt}(y.f) \cap \hat{r}) \land (\textit{Obj}_2 \text{ from pt}(y))$

$z = y.f$

$z \mapsto \textit{Obj}_1 \land (\textit{Obj}_1 \text{ from } \hat{r})$
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. $\hat{a} > \hat{b} \land \hat{a} < \hat{b}$

Simultaneous points-to: e.g. $\hat{\circ} \cdot f \leftrightarrow \hat{u} \ast \hat{\circ} \cdot f \leftrightarrow \hat{v} \land \hat{u} \neq \hat{v}$

From no allocation site: e.g. $\hat{\circ}$ from $\emptyset$

\[
\forall \left( y \mapsto \widehat{Obj}_2 \ast (\widehat{Obj}_2 \cdot f \mapsto \widehat{Obj}_1) \land \left( \widehat{Obj}_1 \text{ from pt}(y.f) \cap \hat{r} \right) \land \left( \widehat{Obj}_2 \text{ from pt}(y) \right) \right)
\]

\[
x \cdot f = p
\]

\[
(y \mapsto \widehat{Obj}_2 \ast (\widehat{Obj}_2 \cdot f \mapsto \widehat{Obj}_1) \land \left( \widehat{Obj}_1 \text{ from pt}(y.f) \cap \hat{r} \right) \land \left( \widehat{Obj}_2 \text{ from pt}(y) \right) \)
\]

\[
z = y \cdot f
\]

\[
z \mapsto \widehat{Obj}_1 \land \left( \widehat{Obj}_1 \text{ from } \hat{r} \right)
\]

Monday, July 8, 13
Can refute if we derive \textbf{false} via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Simultaneous points-to: e.g. \( \hat{\varnothing} \cdot f \leftrightarrow \hat{u} \ast \hat{\varnothing} \cdot f \leftrightarrow \hat{v} \land \hat{u} \neq \hat{v} \)

From no allocation site: e.g. \( \hat{\varnothing} \text{ from } \emptyset \)

\( x \text{ and } y \text{ not aliased, query unchanged} \)

\[ \forall (y \mapsto \overbrace{\text{Obj}_2}^* \ast (\overbrace{\text{Obj}_2 \cdot f} \mapsto \overbrace{\text{Obj}_1}^\land \overbrace{\text{Obj}_1 \text{ from } pt(y.f) \cap \hat{r}}^\land \overbrace{\text{Obj}_2 \text{ from } pt(y)}^\land) \) \]

\( x \cdot f = p \)

\[ (y \mapsto \overbrace{\text{Obj}_2}^* \ast (\overbrace{\text{Obj}_2 \cdot f} \mapsto \overbrace{\text{Obj}_1}^\land \overbrace{\text{Obj}_1 \text{ from } pt(y.f) \cap \hat{r}}^\land \overbrace{\text{Obj}_2 \text{ from } pt(y)}^\land) \) \]

\( z = y \cdot f \)

\[ z \mapsto \overbrace{\text{Obj}_1}^\land \overbrace{\text{Obj}_1 \text{ from } \hat{r}}^\land \]
Can refute if we derive false via:

* Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)
* Simultaneous points-to: e.g. \( \hat{o} \cdot f \leftrightarrow \hat{u} \land \hat{o} \cdot f \leftrightarrow \hat{v} \land \hat{u} \neq \hat{v} \)
* From no allocation site: e.g. \( \hat{o} \) from \( \emptyset \)

\[
(y \mapsto \overline{Obj}_2) \ast (x \mapsto \overline{Obj}_2) \ast (p \mapsto \overline{Obj}_1) \land \\
(Obj_1 \text{ from } pt(p) \cap pt(y.f) \cap \hat{\rho}) \land (Obj_2 \text{ from } pt(x) \cap pt(y)) \lor \\
(y \mapsto \overline{Obj}_2) \ast (\overline{Obj}_2 \cdot f \mapsto \overline{Obj}_1) \land \\
(Obj_1 \text{ from } pt(y.f) \cap \hat{\rho}) \land (Obj_2 \text{ from } pt(y))
\]

\[x \cdot f = p\]

\[
(y \mapsto \overline{Obj}_2) \ast (\overline{Obj}_2 \cdot f \mapsto \overline{Obj}_1) \land \\
(Obj_1 \text{ from } pt(y.f) \cap \hat{\rho}) \land (Obj_2 \text{ from } pt(y))
\]

\[z = y \cdot f\]

\[
z \mapsto \overline{Obj}_1 \land (\overline{Obj}_1 \text{ from } \hat{\rho})
\]
Can refute if we derive false via:

Unsatisfiable path constraint: e.g. \( \hat{a} > \hat{b} \land \hat{a} < \hat{b} \)

Simultaneous points-to: e.g. \( \hat{o} \cdot f \mapsto \hat{u} \land \hat{d} \cdot f \mapsto \hat{v} \land \hat{u} \neq \hat{v} \)

From no allocation site: e.g. \( \hat{o} \) from \( \emptyset \)

\[
(y \mapsto \text{Obj}_2) \ast (x \mapsto \text{Obj}_2) \ast (p \mapsto \text{Obj}_1) \land \\
\left( \text{Obj}_1 \text{ from pt}(p) \cap \text{pt}(y.f) \cap \hat{r} \right) \land \\
\text{Obj}_2 \text{ from pt}(x) \cap \text{pt}(y))
\]

\[
\lor \\
(y \mapsto \text{Obj}_2) \ast (\text{Obj}_2.f \mapsto \text{Obj}_1) \land \\
\left( \text{Obj}_1 \text{ from pt}(y.f) \cap \hat{r} \right) \land (\text{Obj}_2 \text{ from pt}(y))
\]

\[
x.f = p
\]

\[
(y \mapsto \text{Obj}_2) \ast (\text{Obj}_2.f \mapsto \text{Obj}_1) \land \\
\left( \text{Obj}_1 \text{ from pt}(y.f) \cap \hat{r} \right) \land (\text{Obj}_2 \text{ from pt}(y))
\]

\[
z = y.f
\]

\[
z \mapsto \text{Obj}_1 \land (\text{Obj}_1 \text{ from } \hat{r})
\]
Generalized aliasing check.
Can refute immediately if $\text{pt}(x) \cap \text{pt}(y) = \emptyset$,
or later by further restricting from set.

From no allocation site: e.g. $\hat{\delta}$ from $\emptyset$

\[
(y \mapsto \tilde{\text{Obj}}_2) \ast (x \mapsto \tilde{\text{Obj}}_2) \ast (p \mapsto \tilde{\text{Obj}}_1) \land \\
(\text{Obj}_1 \text{ from pt}(p) \cap \text{pt}(y.f) \cap \hat{\mathfrak{r}}) \land (\text{Obj}_2 \text{ from pt}(x) \cap \text{pt}(y))
\]

\[
(x \cdot f = p)
\]

\[
(y \mapsto \tilde{\text{Obj}}_2) \ast (\tilde{\text{Obj}}_2 \cdot f \mapsto \tilde{\text{Obj}}_1) \land \\
(\text{Obj}_1 \text{ from pt}(y.f) \cap \hat{\mathfrak{r}}) \land (\text{Obj}_2 \text{ from pt}(y))
\]

\[
z = y \cdot f
\]

\[
z \mapsto \tilde{\text{Obj}}_1 \land (\tilde{\text{Obj}}_1 \text{ from } \hat{\mathfrak{r}})
\]
Experiments

Program

Points-to analysis

Fact database

Lots of alarms

Fixes

Manual triage

Fewer alarms

Thresher triage
Experiments

(1) Is leveraging points-to facts important for scalability?

Program

Points-to analysis

Fact database

Lots of alarms

Manual triage

Fewer alarms

Thresher triage

Fixes

Program

Monday, July 8, 13
(1) Is leveraging points-to facts important for scalability?

(2) Is Thresher effective at filtering false alarms from the points-to analysis?
Effectiveness of leveraging points-to facts
Effectiveness of leveraging points-to facts

• Implemented two alternatives to \textit{from} constraints:
Effectiveness of leveraging points-to facts

- Implemented two alternatives to \texttt{from} constraints:
  
  (1) \textit{fully symbolic} (points-to analysis for aliasing checks, but no \texttt{from} constraints)
Effectiveness of leveraging points-to facts

- Implemented two alternatives to \texttt{from} constraints:
  
  (1) \textit{fully symbolic} (points-to analysis for aliasing checks, but no \texttt{from} constraints)

  (2) \textit{fully explicit} (eager case split on allocation sites in \texttt{from} constraint)
Effectiveness of leveraging points-to facts

- Implemented two alternatives to \texttt{from} constraints:

  (1) \textit{fully symbolic} (points-to analysis for aliasing checks, but no \texttt{from} constraints)

  (2) \textit{fully explicit} (eager case split on allocation sites in \texttt{from} constraint)

Symbolic representation: 1-4X slowdown

Explicit representation: path budget exhausted
Effectiveness of leveraging points-to facts

• Implemented two alternatives to from constraints:

  (1) fully symbolic (points-to analysis for aliasing checks, but no from constraints)

  (2) fully explicit (eager case split on allocation sites in from constraint)

Symbolic representation: 1-4X

Explicit representation: path budget exhausted

Can’t refute edge within 10000 paths = timeout, soundly consider edge not refuted
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
<th>False Alarm %</th>
<th>Filtered %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
<td>18</td>
<td>90</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>72K</td>
<td>311</td>
<td>172</td>
<td>115</td>
<td>1602</td>
<td>17</td>
<td>88</td>
</tr>
<tr>
<td>Program</td>
<td>LOC</td>
<td>PT Alarms</td>
<td>Thresher Refuted</td>
<td>True Alarms</td>
<td>Time (s)</td>
<td>False Alarm %</td>
<td>Filtered %</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
<td>-----------</td>
<td>------------------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------</td>
<td>------------</td>
</tr>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
<td>18</td>
<td>90</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72K</strong></td>
<td><strong>311</strong></td>
<td><strong>172</strong></td>
<td><strong>115</strong></td>
<td><strong>1602</strong></td>
<td><strong>17</strong></td>
<td><strong>88</strong></td>
</tr>
</tbody>
</table>

App LOC only--
Android library ~880K
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72K</strong></td>
<td><strong>311</strong></td>
</tr>
</tbody>
</table>

*(static field, Activity pairs)*
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72K</strong></td>
<td><strong>311</strong></td>
<td><strong>172</strong></td>
</tr>
</tbody>
</table>

Automated triaging
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>72K</td>
<td>311</td>
<td>172</td>
<td>115</td>
</tr>
</tbody>
</table>

Manual triage

Monday, July 8, 13
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72K</strong></td>
<td><strong>311</strong></td>
<td><strong>172</strong></td>
<td><strong>115</strong></td>
<td><strong>1602</strong></td>
</tr>
</tbody>
</table>

Not including points-to time (10-40 s)
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72K</strong></td>
<td><strong>311</strong></td>
<td><strong>172</strong></td>
<td><strong>115</strong></td>
<td><strong>1602</strong></td>
</tr>
</tbody>
</table>

Time is reasonable: coffee to lunch break
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>72K</strong></td>
<td><strong>311</strong></td>
<td><strong>172</strong></td>
<td><strong>115</strong></td>
<td><strong>1602</strong></td>
</tr>
</tbody>
</table>

- LOC not strongly correlated with analysis time
- Time is reasonable: coffee to lunch break

Monday, July 8, 13
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>72K</td>
<td>311</td>
<td>172</td>
<td>115</td>
<td>1602</td>
</tr>
<tr>
<td>Program</td>
<td>LOC</td>
<td>PT Alarms</td>
<td>Thresher Refuted</td>
<td>True Alarms</td>
<td>Time (s)</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
<td>-----------</td>
<td>------------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>72K</td>
<td>311</td>
<td>172</td>
<td>115</td>
<td>1602</td>
</tr>
</tbody>
</table>

Total % of reported alarms that are false alarms
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
<th>False Alarm %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
<td>0</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
<td>100</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>K9Mail</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>

**Total % of reported alarms that are false alarms**

Monday, July 8, 13
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
<th>False Alarm %</th>
<th>Filtered %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>DroidLife</td>
<td>3K</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMSPopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>K9Mail</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>90</td>
</tr>
</tbody>
</table>

**Down from 63% for PT analysis**

**Total % of PT false alarms refuted**

Monday, July 8, 13
<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>PT Alarms</th>
<th>Thresher Refuted</th>
<th>True Alarms</th>
<th>Time (s)</th>
<th>False Alarm %</th>
<th>Filtered %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsePoint</td>
<td>no src</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>95</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>StandupTimer</td>
<td>2K</td>
<td>25</td>
<td>15</td>
<td>0</td>
<td>1068</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>DroidLife</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>SMS PopUp</td>
<td>7K</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>aMetro</td>
<td>20K</td>
<td>54</td>
<td>18</td>
<td>36</td>
<td>18</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>K9Mail</td>
<td>40K</td>
<td>208</td>
<td>130</td>
<td>64</td>
<td>374</td>
<td>18</td>
<td>90</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>72K</td>
<td>311</td>
<td>172</td>
<td>115</td>
<td>1602</td>
<td>17</td>
<td>88</td>
</tr>
</tbody>
</table>

Total % of PT false alarms refuted

**Filters 88% of PT false alarms**

**Down from 63% for PT analysis**
Conclusion
Conclusion

Refute PT analysis false alarms by adding \{path, context\}-sensitivity, strong updates on demand

Lots of alarms

Thresher triage
Conclusion

Refute PT analysis false alarms by adding \{path, context\}-sensitivity, strong updates on demand

from constraints leverage up-front points-to result to make backwards analysis scale

Fact database

Lots of alarms

Thresher triage

Monday, July 8, 13
Conclusion

Refute PT analysis false alarms by adding \{path, context\}-sensitivity, strong updates on demand.

Can filter 88% of PT analysis leak alarms during lunch break.
Advertising

Thresher tool available
github.com/cuplv/thresher

pl.cs.colorado.edu