Race Detection for Android Applications

Pallavi Maiya, **Aditya Kanade** (Indian Institute of Science)

Rupak Majumdar (Max Planck Institute for Software Systems)

PLDI 2014

Popularity of Android Applications

Million+ Android apps in the market

Billions of downloads

Our Contributions

- Formalizing concurrency semantics of Android applications
- Encoding happens-before relation for them
  - Algorithm to detect and categorize data races
  - Environment modeling to reduce false positives
- A dynamic tool to detect data races (DroidRacer)
  - Performs systematic testing
  - Identified potential races in popular applications like

Android Concurrency Model

Multithreading constructs: threads, synchronization

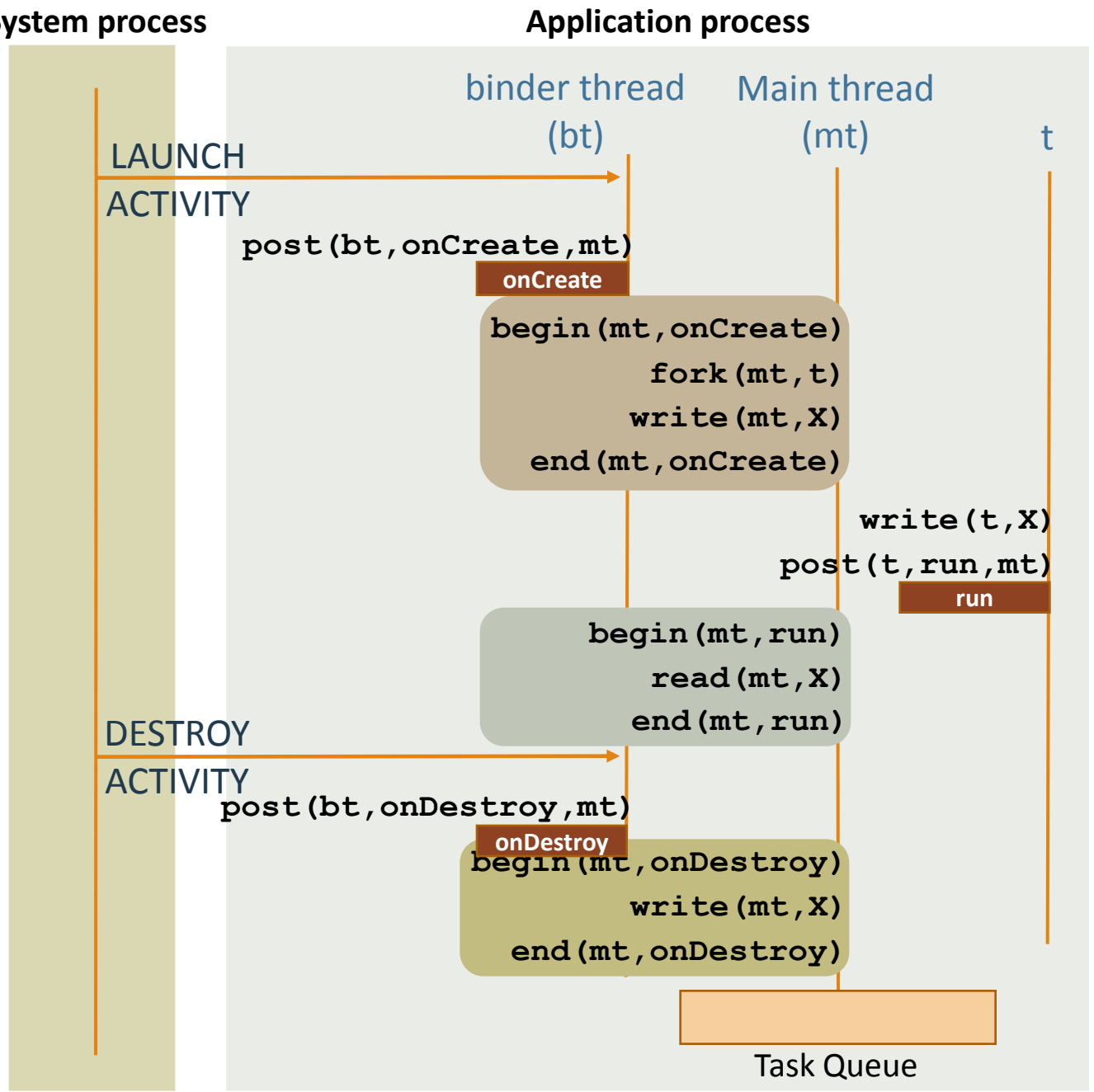
- Dynamic thread creation (fork)
- Lock – unlock, join, notify – wait

Asynchrony constructs: task queues, posting asynchronous tasks

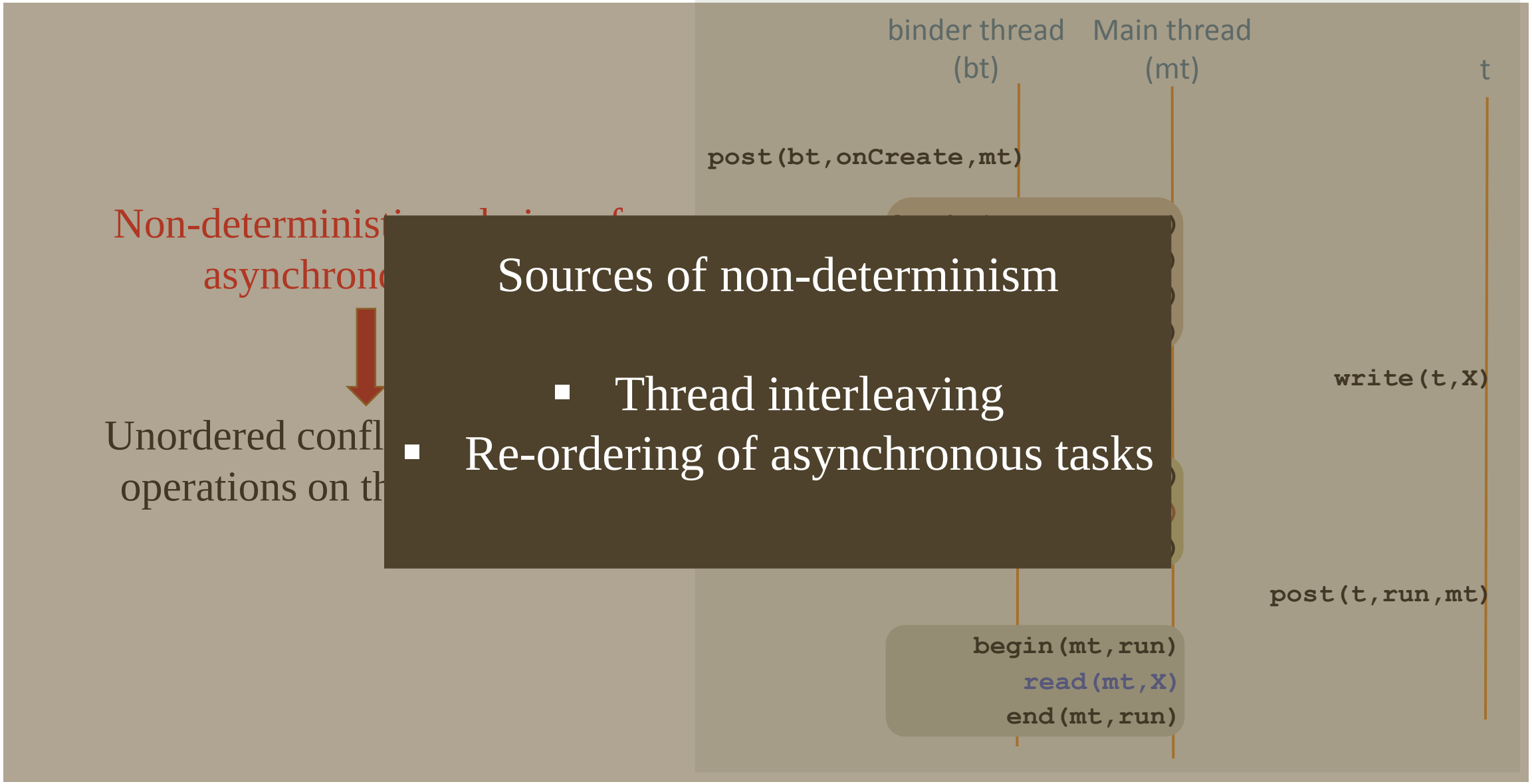
- Threads may be associated with task queues
- Any thread can post an asynchronous task to a task queue
  - Tasks may be associated with timeouts or posted to front-of-queue
- Tasks executed in FIFO order
- Atomic execution of asynchronous tasks

Concurrency Semantics

```
public class MainActivity extends Activity
{
  int X;
  protected void onCreate( ){
    Runnable r  =  new Runnable( ){
      public void run( ){
        X  =  2;
        runOnUiThread(new Runnable( ){
          public void run( ){
            System.out.println(X);
          }
        });
      }
    };
    Thread t  =  new Thread(r);
    t.start( );
    X  =  1;
  }
  protected void onDestroy( ){
    X  =  -1;
  }
}
```



Single-threaded Race



Race Detection : Happens-before Reasoning

Multi-threading **without Asynchrony** **with Asynchrony** (e.g. Android)

Thread-local ordering

- Total order between all operations on a thread (program order)
- Total order only between operations on a thread with no task queue
- Total order between all operations in the same asynchronous task

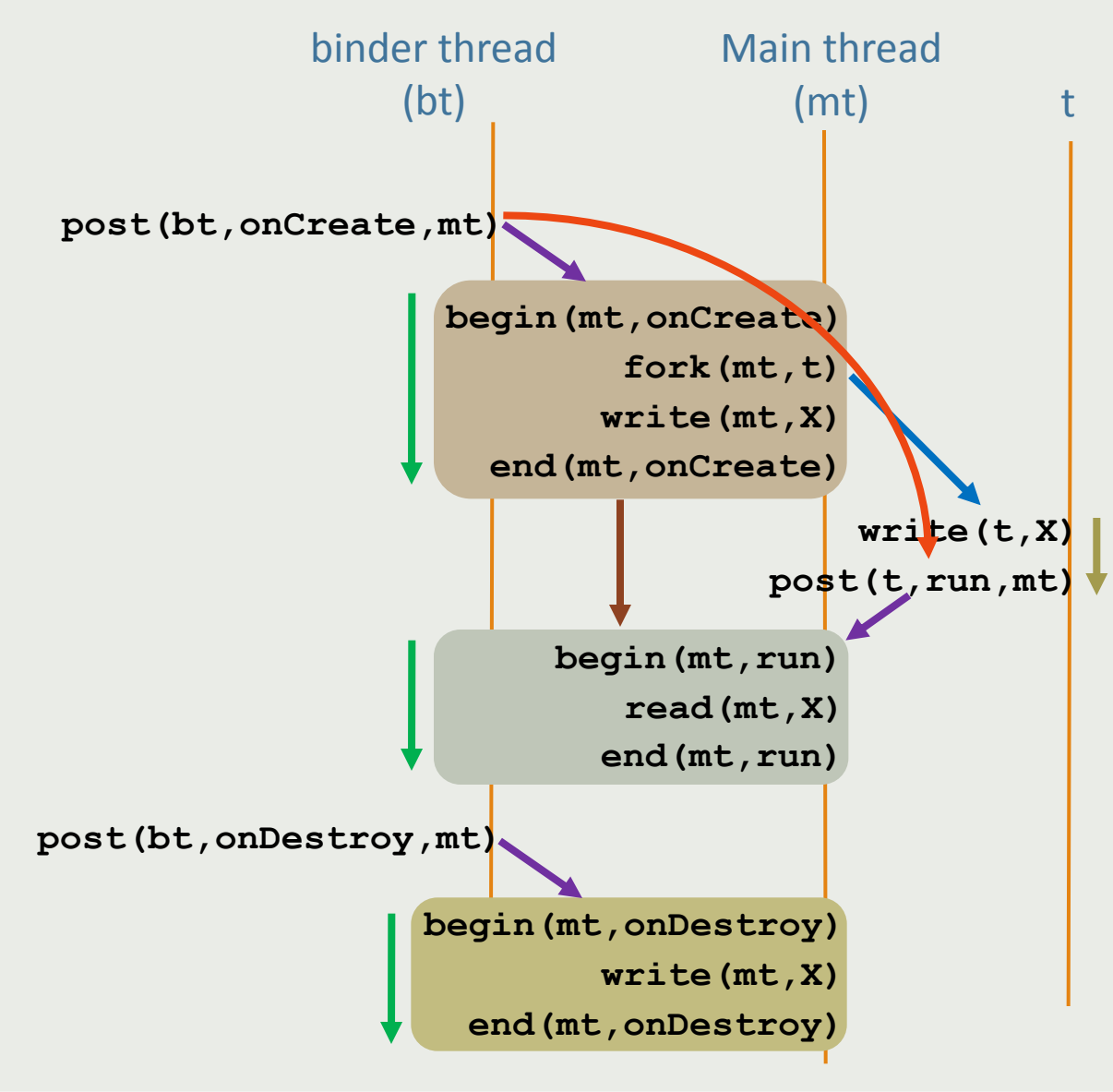
(+ additional rules)

Inter-thread ordering

- FORK happens-before init of newly forked thread
- Thread exits before JOINing to another thread
- UNLOCK happens-before a subsequent LOCK on the same monitor

(+ additional rules)

Happens-before Relation for Android Applications



ASYNC-PO NO-Q-PO

FORK JOIN

POST

FIFO

NO-PRE

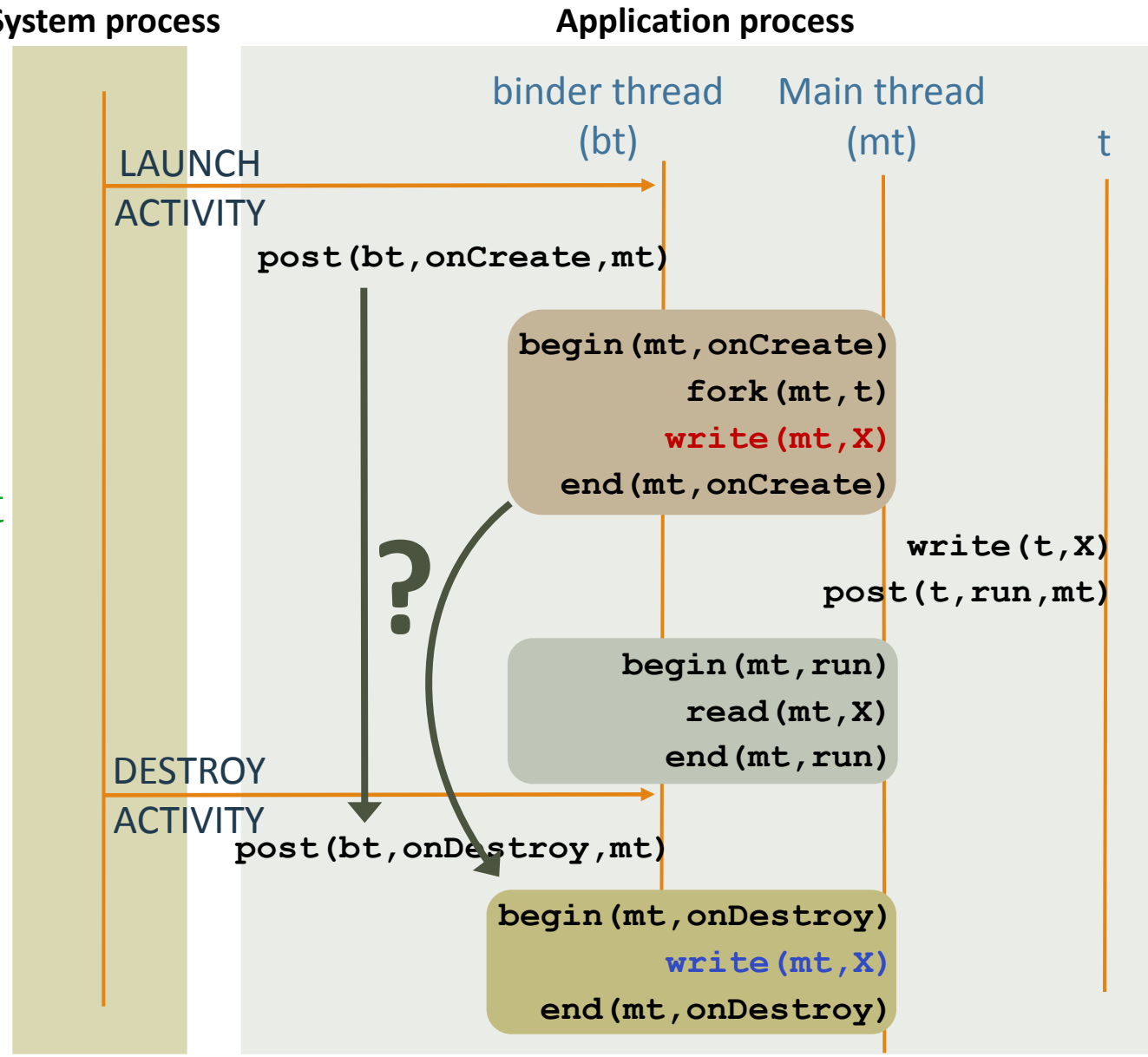
LOCK

TRANSITIVITY

Environment Modeling

✗ Track system process and IPC

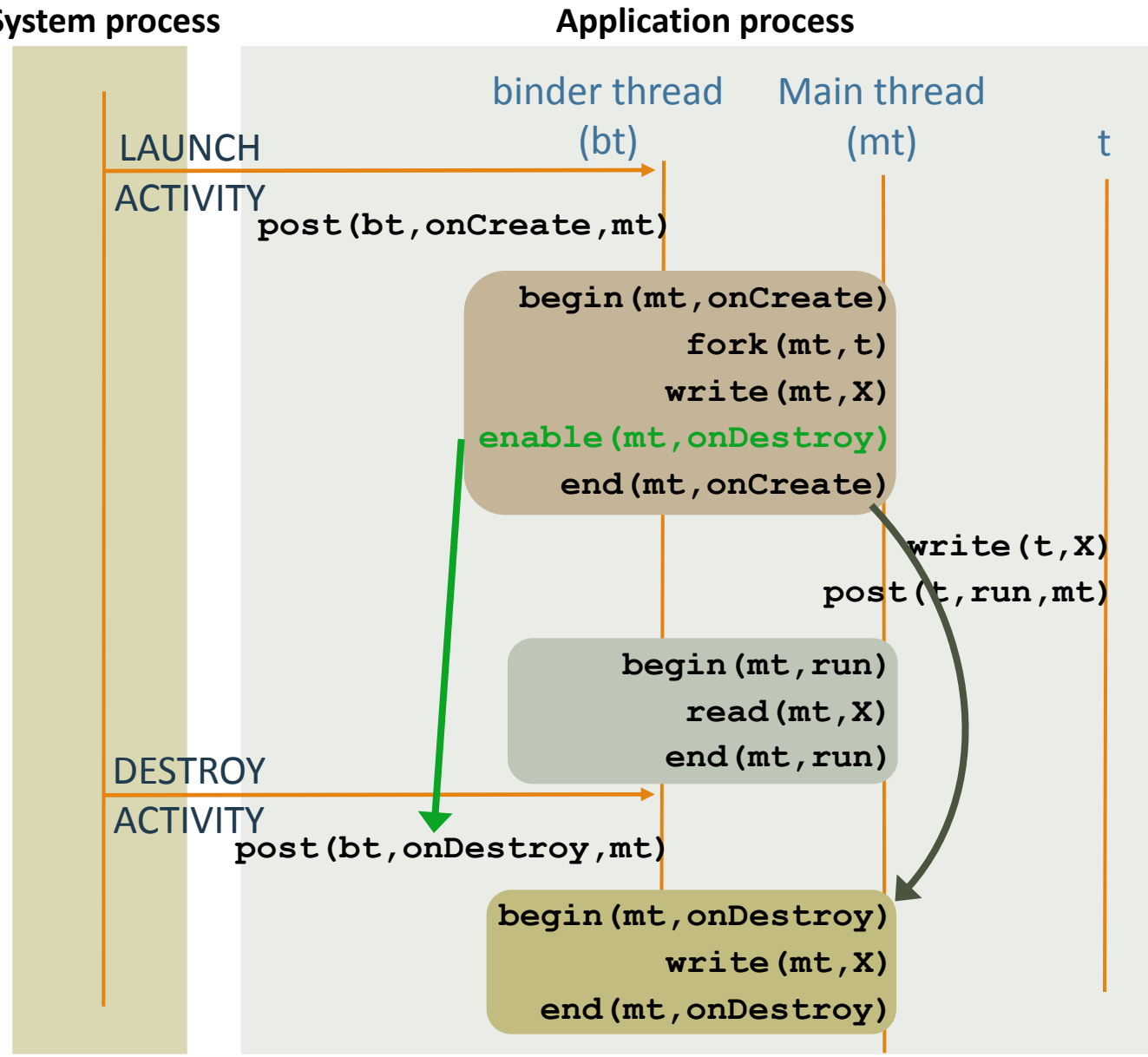
✓ Model the effect of the environment in ordering of operations



Environment Modeling

Ordering due to environment modeled
using enable operation

enable(_, m) HB post (_, m)



DroidRacer Algorithm

- Acyclic graph representation of happens-before constraints
  - Nodes: operations in trace Edges: happens-before relation
  - Saturate the graph with happens-before rules
  - Report conflicting memory operations with no happens-before relation as race
- Debugging assistance
  - Method stack, high level events
- Classification of reported data races

Classification of Data Races

Type of data race		Sources of non-determinism
Multi-threaded race		Thread interleaving
Single-threaded race	**Co-enabled**	High level events causing the conflicting operations are unordered
	Cross-posted	Non-deterministic interleaving of two threads posting tasks to the same target thread
	Delayed	At least one of the conflicting operations is due to a task with timeout. This breaks FIFO.

DroidRacer – Dynamic Data Race Detection Tool

UI Explorer – Systematic Testing

- Depth first traversal with backtracking
- Supports click, long press, data input, rotate screen ….

Trace Generator

- Logs concurrency constructs and read-writes
- Logs debug information

Race Detector

- Happens-before graph constructed on generated trace
- Categorization of data races

Android core library and Dalvik virtual machine of Android 4.0 instrumented.

Experimental Evaluation – Trace Statistics*

Applications	Trace length	Fields	Threads (w/o Q)	Threads (w/ Q)	Async. tasks
Aard Dictionary	1k	189	2	1	58
Music Player	5k	521	3	2	62
My Tracks	7k	573	11	**7**	164
Messenger	10k	845	11	4	99
Tomdroid Notes	10k	413	3	1	348
FBReader	10k	322	14	1	119
Browser	19k	963	13	4	103
OpenSudoku	25k	334	5	1	45
K-9 Mail	30k	1296	7	2	**689**
SGTPuzzles	39k	566	4	1	80
Remind Me	10k	348	3	1	176
Twitter	17k	1362	21	5	97
Adobe Reader	34k	1267	17	4	226
Facebook	52k	801	16	3	16
Flipkart	**157k**	**2065**	**36**	3	105

* Representative trace for each of the tested application

Experimental Evaluation – Data Races in given Trace

Applications	Multi-threaded	Cross-posted	Co-enabled	Delayed
Aard Dictionary	1 (1)	0	0	0
Music Player	0	17 (4)	11 (10)	4 (0)
My Tracks	1 (0)	2 (1)	1 (0)	0
Messenger	1 (1)	15 (5)	4 (3)	2 (2)
Tomdroid Notes	0	5 (2)	1 (0)	0
FBReader	1 (0)	22 (22)	14 (4)	0
Browser	2 (1)	64 (2)	0	0
OpenSudoku	1 (0)	1 (0)	0	0
K-9 Mail	9 (2)	0	1 (0)	0
SGTPuzzles	11 (10)	21 (8)	0	0
TOTAL	**27 (15)**	**147 (44)**	**32 (17)**	**6 (2)**
Remind Me	0	21	33	0
Twitter	0	20	7	4
Adobe Reader	34	73	0	9
Facebook	12	10	0	0
Flipkart	12	152	84	30

X (Y) :
Races reported
(True Positives)

True positives: 37%

Bad behaviour

Related Work

Race detection for multi-threaded programs

- Savage et al., TOCS ’97 (locksets)
- FastTrack by Flanagan and Freund, PLDI ’09 (vector-clocks)
- Pozniansky and Schuster, Concurr. Comput.: Pract. Exper. ’07 (hybrid technique)

Race detection for single-threaded event-driven programs

- Petrov et al., PLDI ’12
- Raychev et al., OOPSLA ’13
- Zheng et al., WWW ’11

Race detection for multi-threaded and asynchronous programs

- Kahlon et al., FSE ’09 (for C programs – only reports multithreaded races)
- Hsio et al., PLDI ’14 (for Android applications)

Conclusions

- Formalization of Android concurrency model and happens-before rules to capture causality in this model.
- Implemented DroidRacer, a dynamic data race detection tool for Android applications.

Future Work

- Transitive closure slows down on long traces – device faster algorithms to infer happens-before relation (e.g., vector clocks)

- Reduce false positives : ad-hoc synchronization, concurrency operations by native threads, better environment model

DroidRacer webpage

http://www.iisc-seal.net/droidracer

Thank You

Backup Slides

Happens-before Relation for Android Applications

Thread-local rules (HB-S): ordering between operations on the same thread.

t : thread m1, m2 : asynchronous tasks

[NO-Q-PO] Total order between all operations only on threads without task queues.

[ASYNC-PO] Total order between all operations in asynchronous task.

[POST-ST] post(t, m1, t) HB-S begin(t, m1)

[FIFO] If post(_, m1, t) HB-S post(_, m2, t) then task m1 is executed before task m2.

[NO-PRE] a: an operation in task m1

If a HB-S begin(t, m2) then end(t, m1) HB-S begin(t, m2)

Transitive closure

Happens-before Relation for Android Applications

Inter-thread rules (HB-M): ordering between operations on different threads.

t1, t2 : thread m : asynchronous task

[FORK] Fork operation HB-M init of newly forked thread.

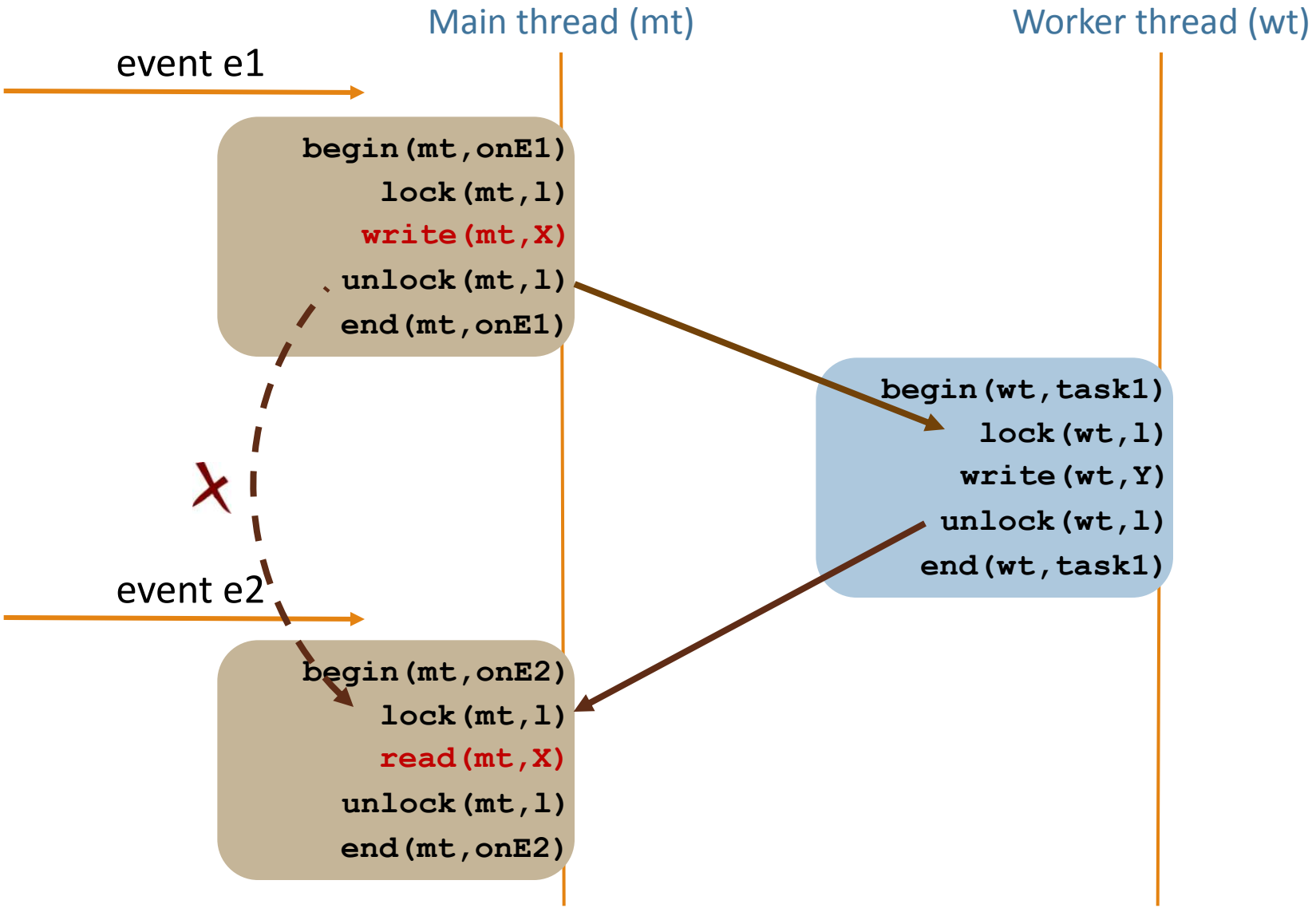
[JOIN] A thread completes execution before joining to another thread.

[LOCK] Unlock HB-M subsequent lock on the same monitor.

[POST-MT] post(t1, m, t2) HB-M begin(t2, m)

Transitive closure (using both HB-M and HB-S)

Naïve transitive closure misses races!



Environment Modeling - Activity