Drinking from Both Glasses: Combining Pessimistic and Optimistic Tracking of Cross-Thread Dependences

MAN CAO

MINJIA ZHANG

ARITRA SENGUPTA

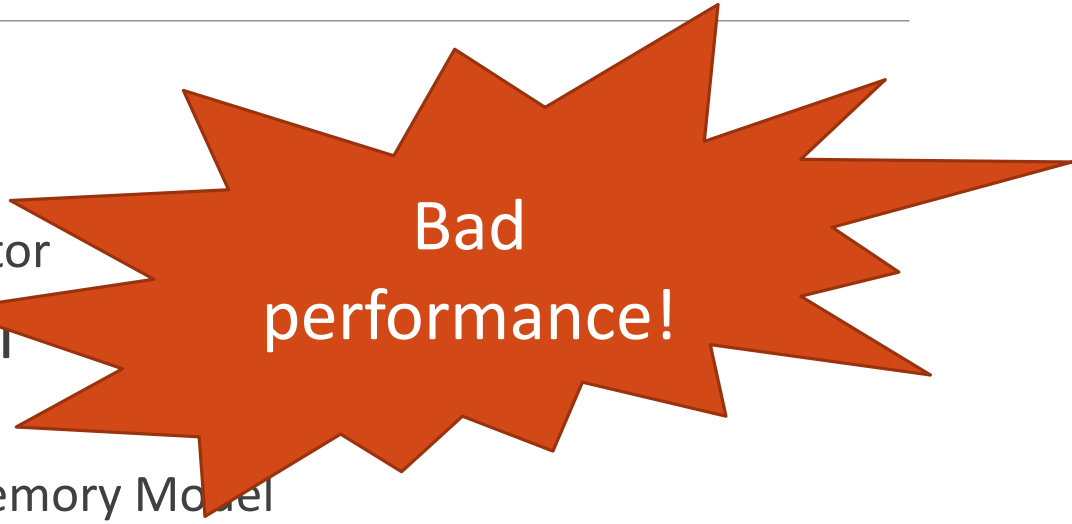
MICHAEL D. BOND

Dynamic Analyses for Parallel Programs

- Error detection
  - Data Race Detector
  - Atomicity Violation Detector
- Programming model
  - Transactional Memory
  - Enforcement of Strong Memory Model
- Debugging
  - Record & Replay
  - Deterministic Execution

Dynamic Analyses for Parallel Programs

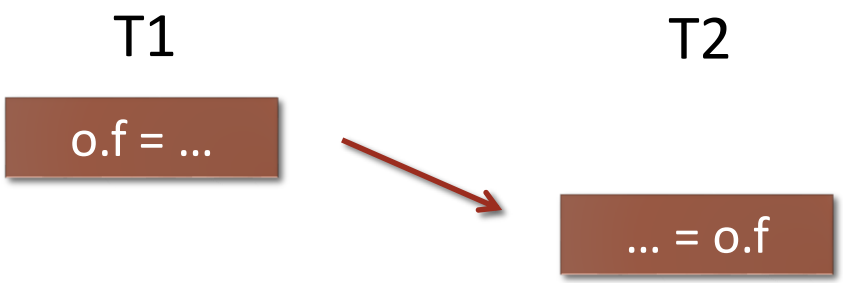
- Error detection
  - Data Race Detector
  - Atomicity Violation Detector
- Programming model
  - Transactional Memory
  - Enforcement of Strong Memory Model
- Debugging
  - Record & Replay
  - Deterministic Execution



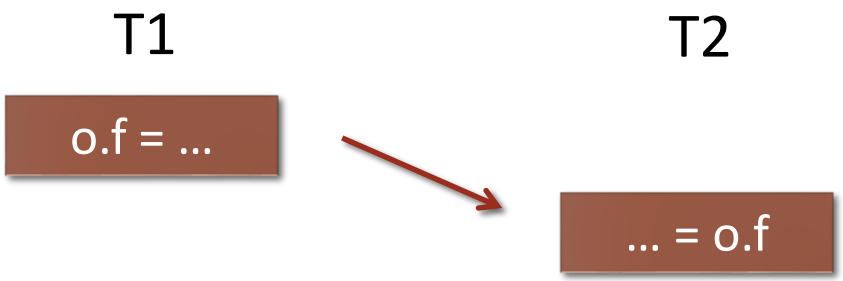
Dynamic Analyses for Parallel Programs

- Error detection
  - Data Race Detector
  - Atomicity Violation Detector
- Programming model
  - Transactional Memory
  - Enforcement of Strong Memory Model
- Debugging
  - Record & Replay
  - Deterministic Execution

Cross-thread dependences

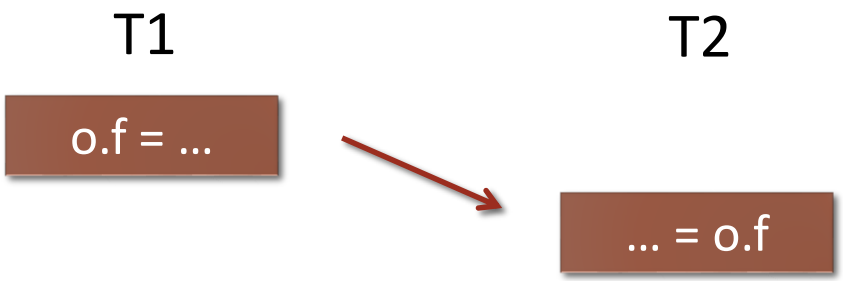


Cross-thread dependences



Tracking cross-thread dependences

Cross-thread dependences



Tracking cross-thread dependences

- Detecting
- Controlling

Outline

- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Pessimistic Tracking

Per-object metadata: o.state

last writer/reader thread

Pessimistic Tracking

Per-object metadata: o.state

last writer/reader thread

At each object access:

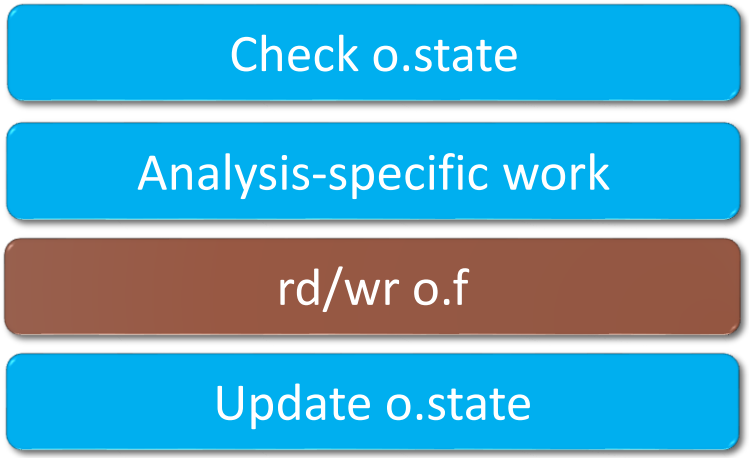
rd/wr o.f

Pessimistic Tracking

Per-object metadata: o.state

last writer/reader thread

At each object access:

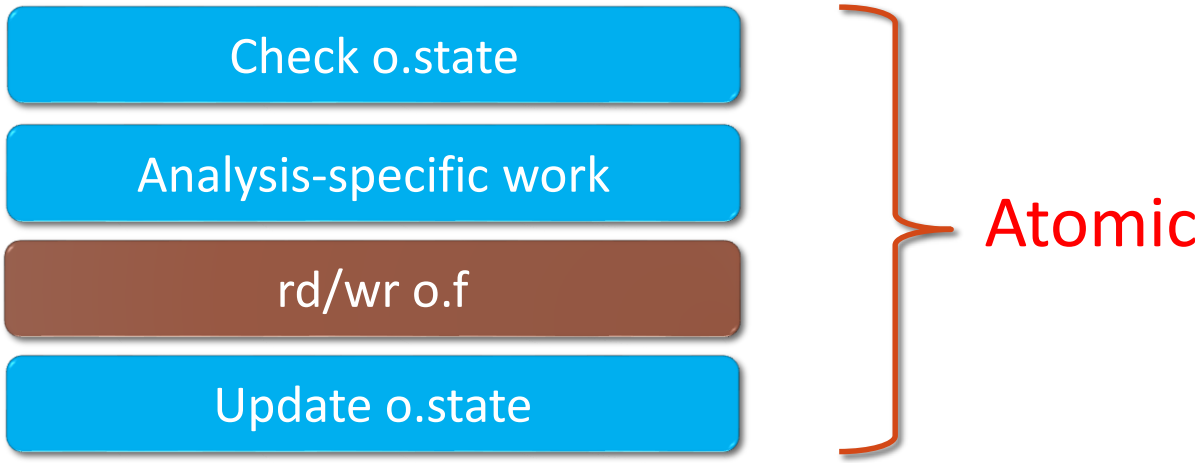


Pessimistic Tracking

Per-object metadata: o.state

last writer/reader thread

At each object access:

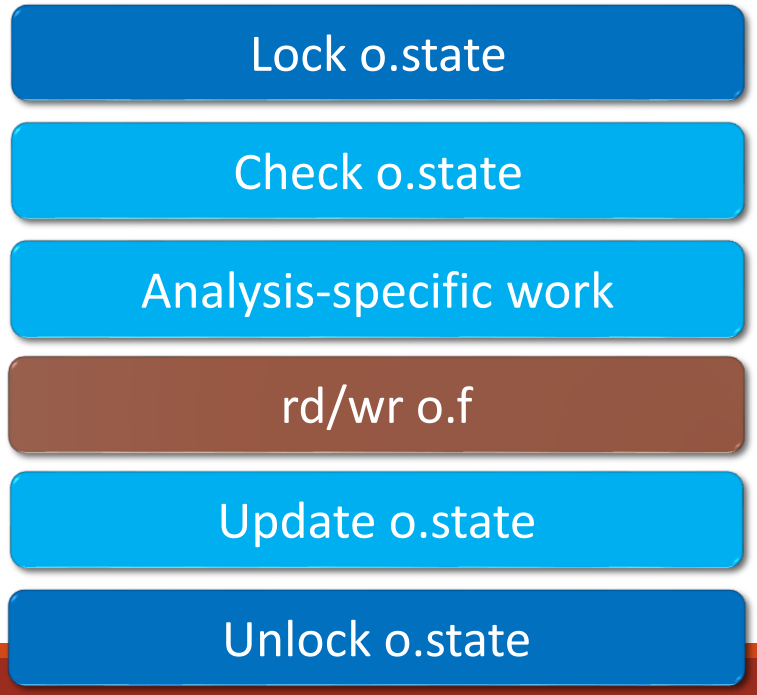


Pessimistic Tracking

Per-object metadata: o.state

last writer/reader thread

At each object access:

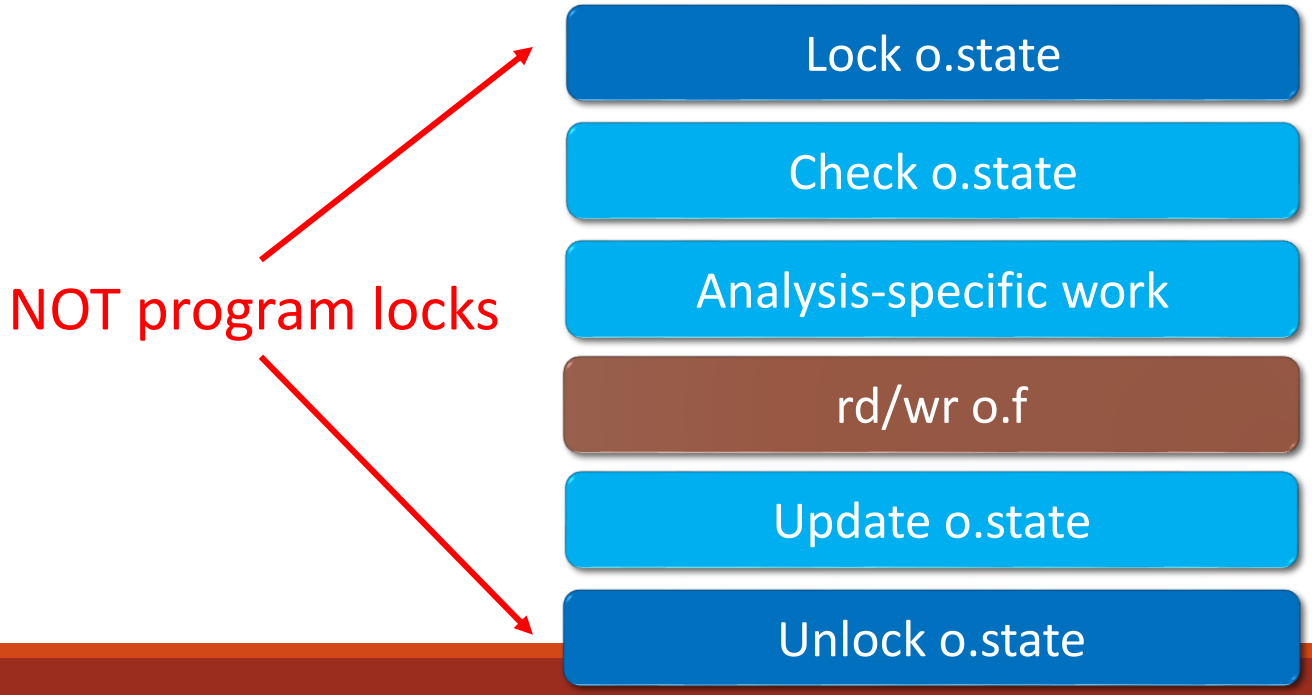


Pessimistic Tracking

Per-object metadata: o.state

last writer/reader thread

At each object access:



Pessimistic Tracking

Lock o.state

…

wr o.f

Unlock o.state

Lock o.state

…

rd o.f

Unlock o.state

Pessimistic Tracking

Lock o.state

…

wr o.f

Unlock o.state

Synchronization and write metadata

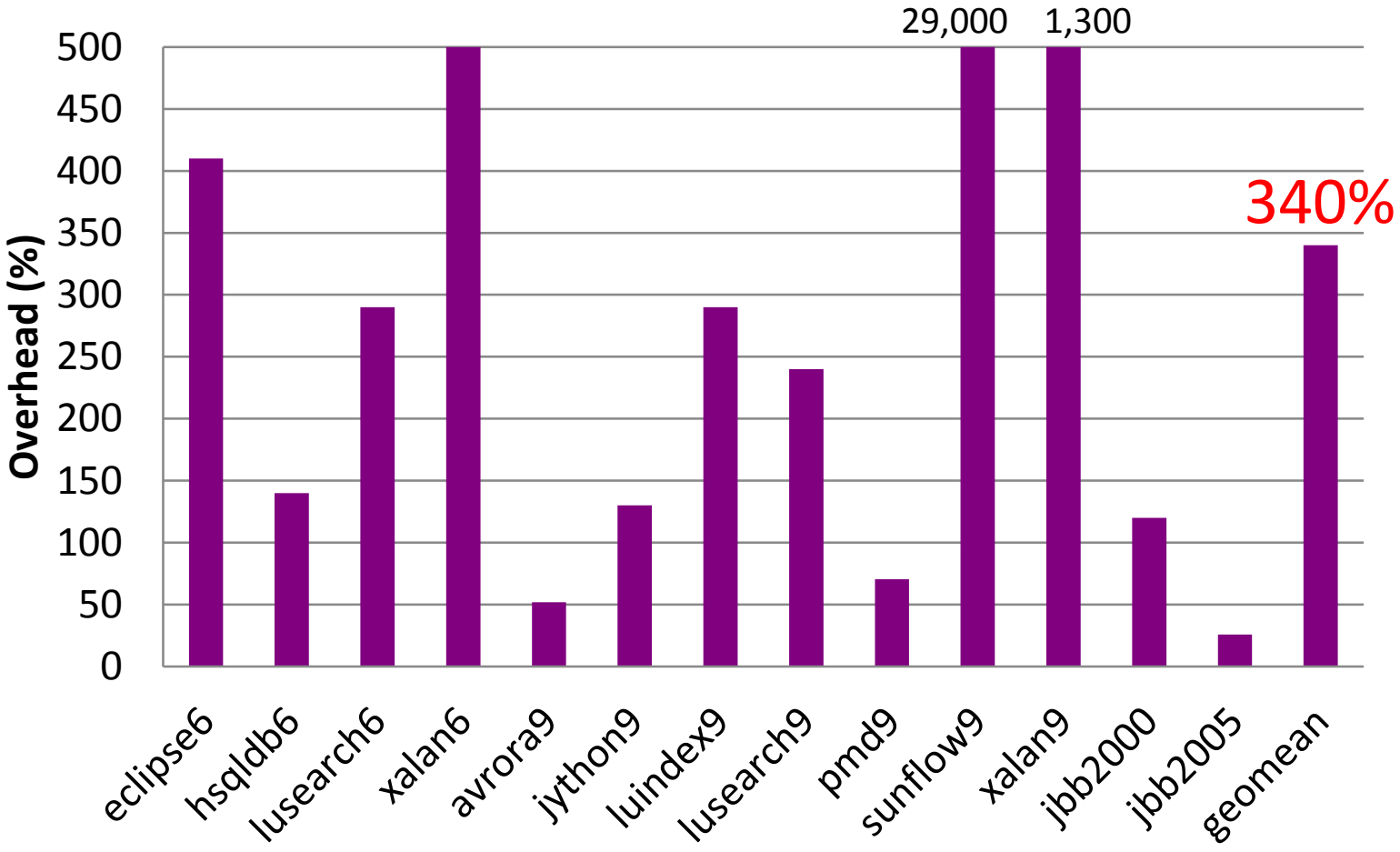
Lock o.state

…

rd o.f

Unlock o.state

Performance of Pessimistic Tracking Alone



Outline

- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Optimistic Tracking

- Biased reader-writer lock for o.state

Optimistic Tracking

- Biased reader-writer lock for o.state
- Avoid synchronization for non-conflicting accesses

Optimistic Tracking

- Biased reader-writer lock for o.state
- Avoid synchronization for non-conflicting accesses
- Heavyweight coordination for conflicting accesses

Optimistic Tracking

T1 T2

o.state: WrExT1

write check

wr o.f

Optimistic Tracking

T1

T2

o.state: WrExT1

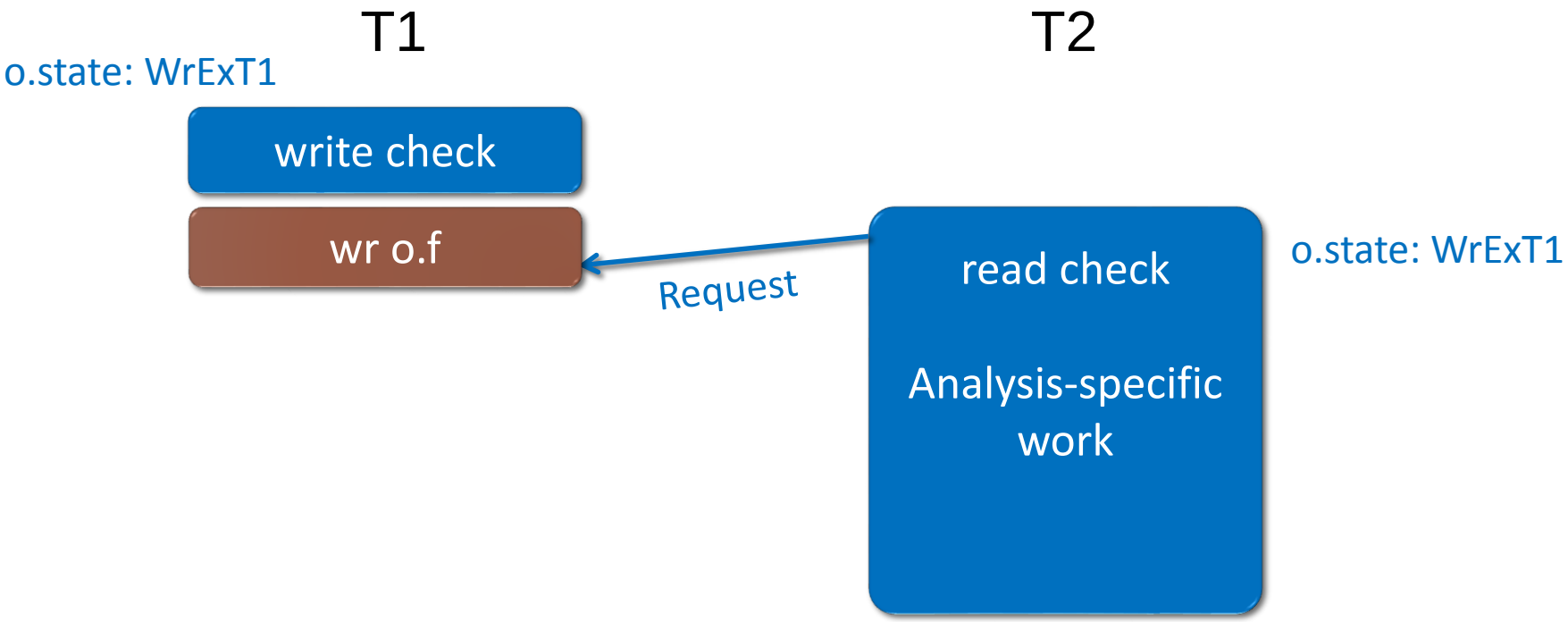
write check

wr o.f

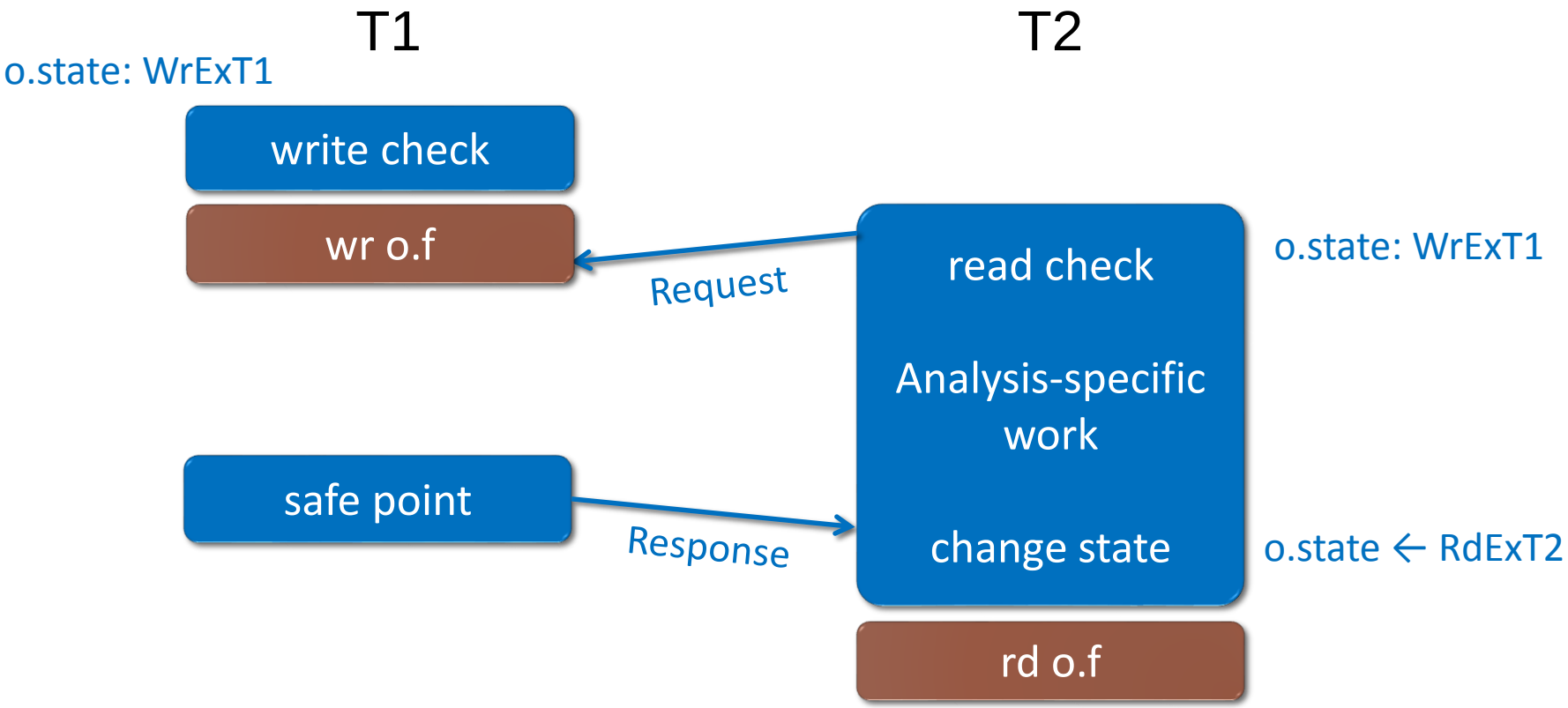
read check

o.state: WrExT1

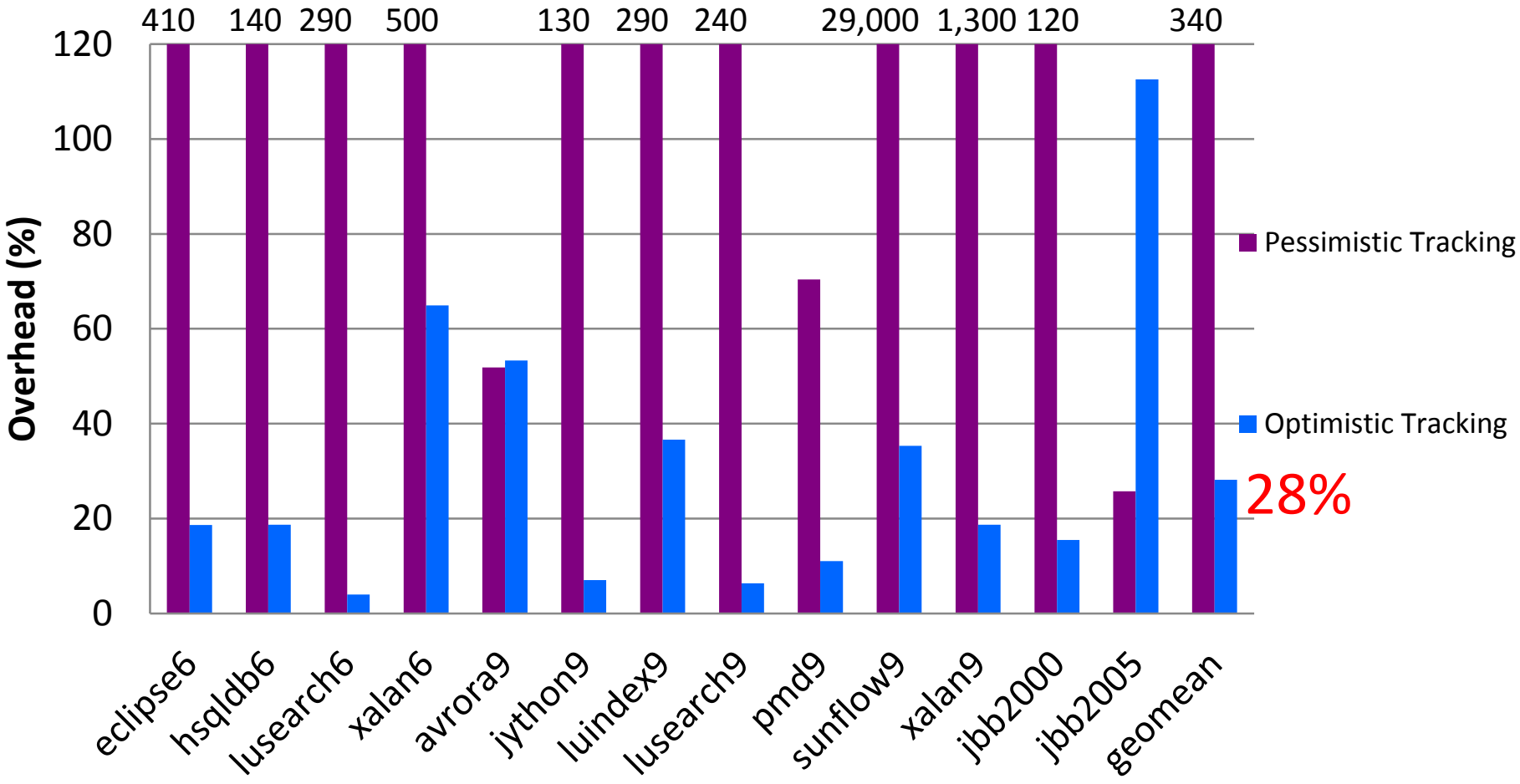
Optimistic Tracking



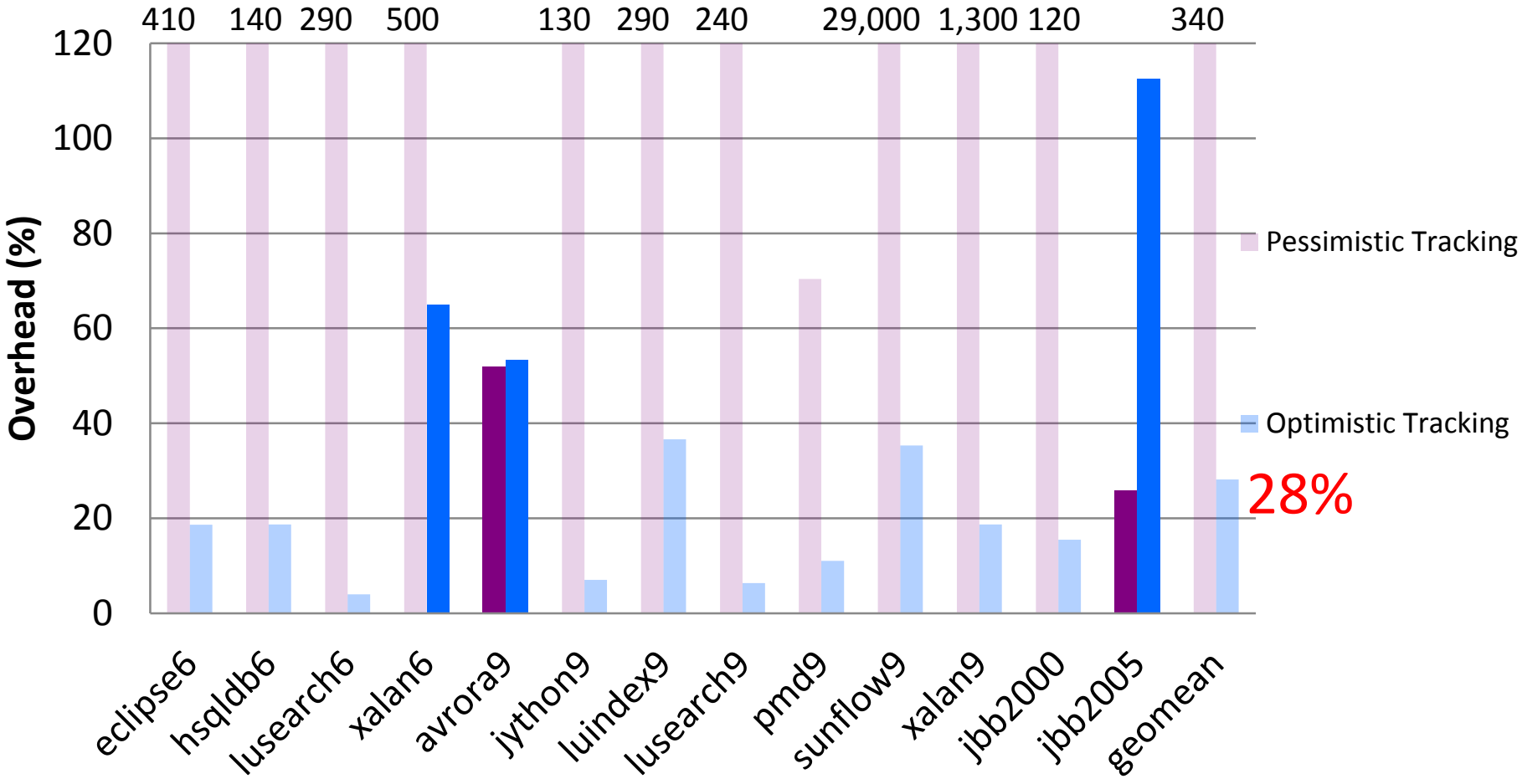
Optimistic Tracking



Performance of Optimistic Tracking Alone



Performance of Optimistic Tracking Alone



Cost of Different Tracking

Pessimistic	Optimistic	
	Same state	Coordination
150	47	9200

- In CPU cycles
- Averaged across all programs

Optimistic tracking performs best if there are few conflicting accesses.

Pessimistic tracking is cheaper for conflicting accesses.

Drink from both glasses?

Goal

- Optimistic tracking for most non-conflicting accesses
- Pessimistic tracking for most conflicting accesses

Outline

- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Outline

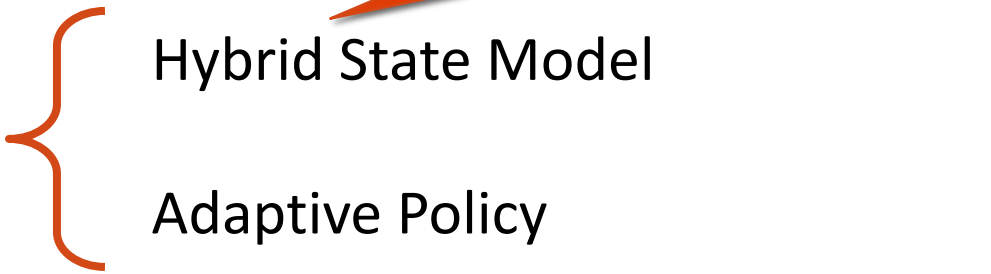
- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Hybrid State Model

Adaptive Policy

Outline

- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

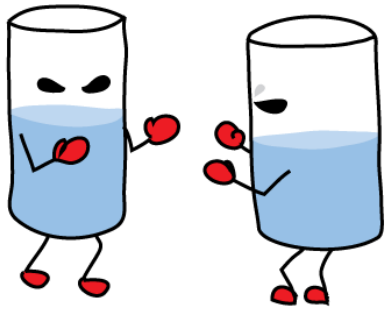


Outline

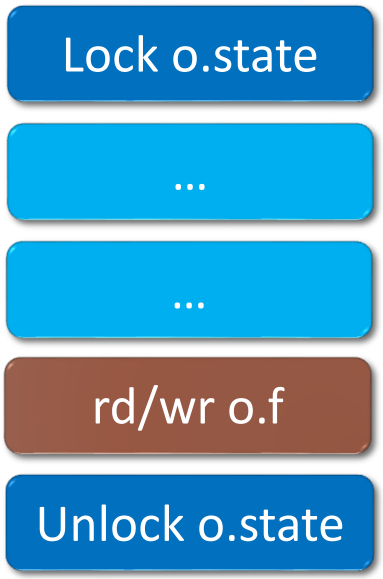
- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Hybrid State Model
Deferred Unlocking
Adaptive Policy

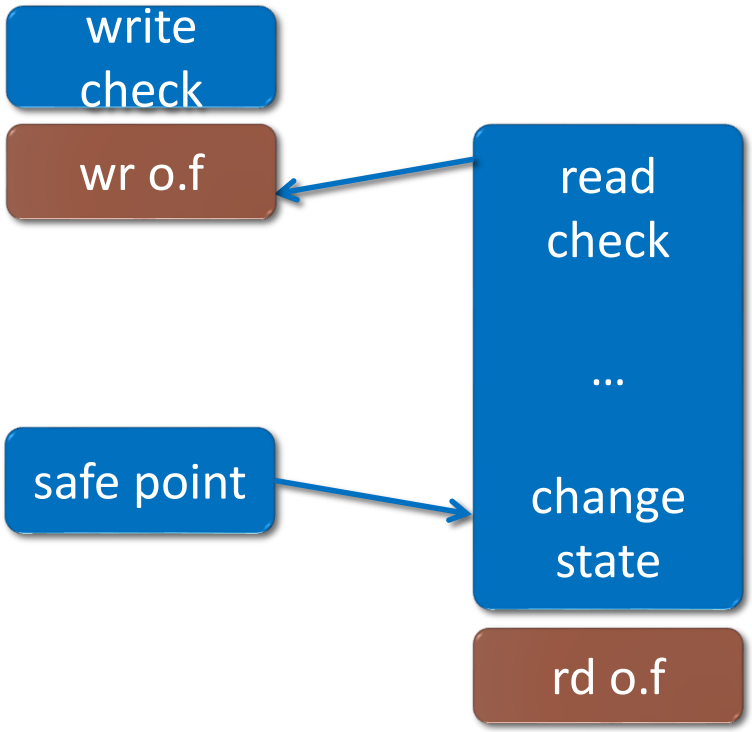
Pessimistic-Optimistic Mismatch



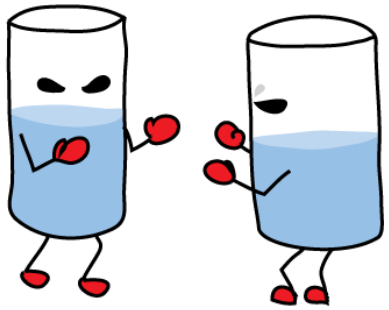
Pessimistic Tracking



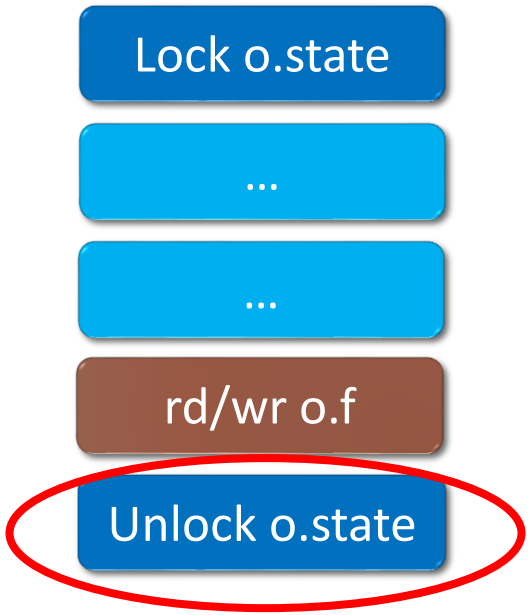
Optimistic Tracking



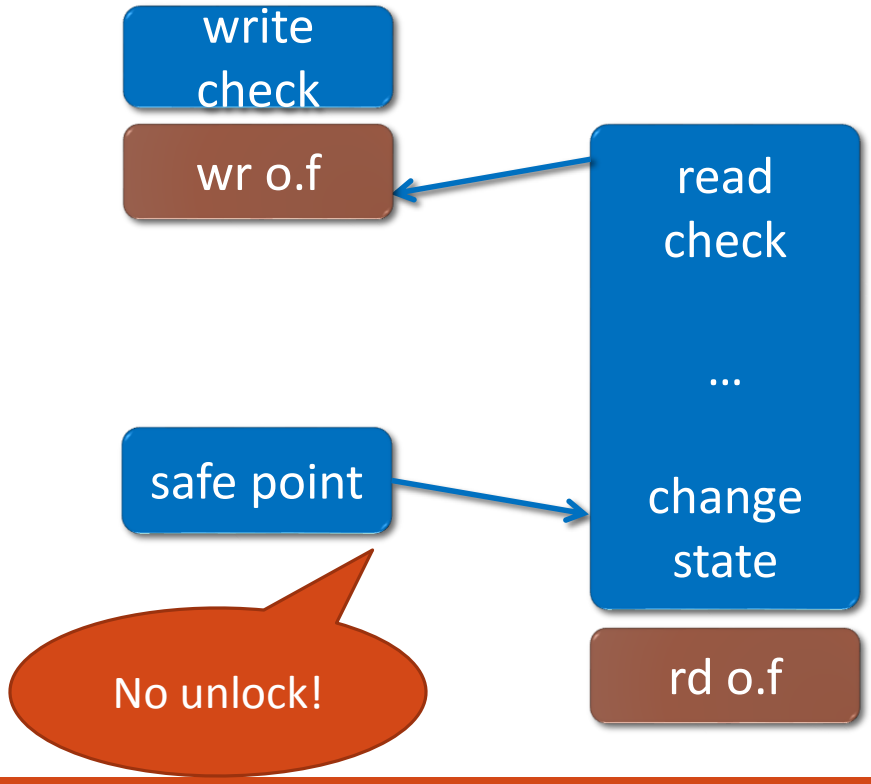
Pessimistic-Optimistic Mismatch (#1)



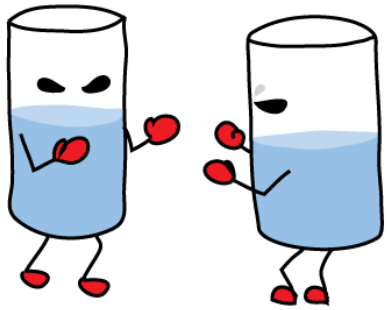
Pessimistic Tracking



Optimistic Tracking

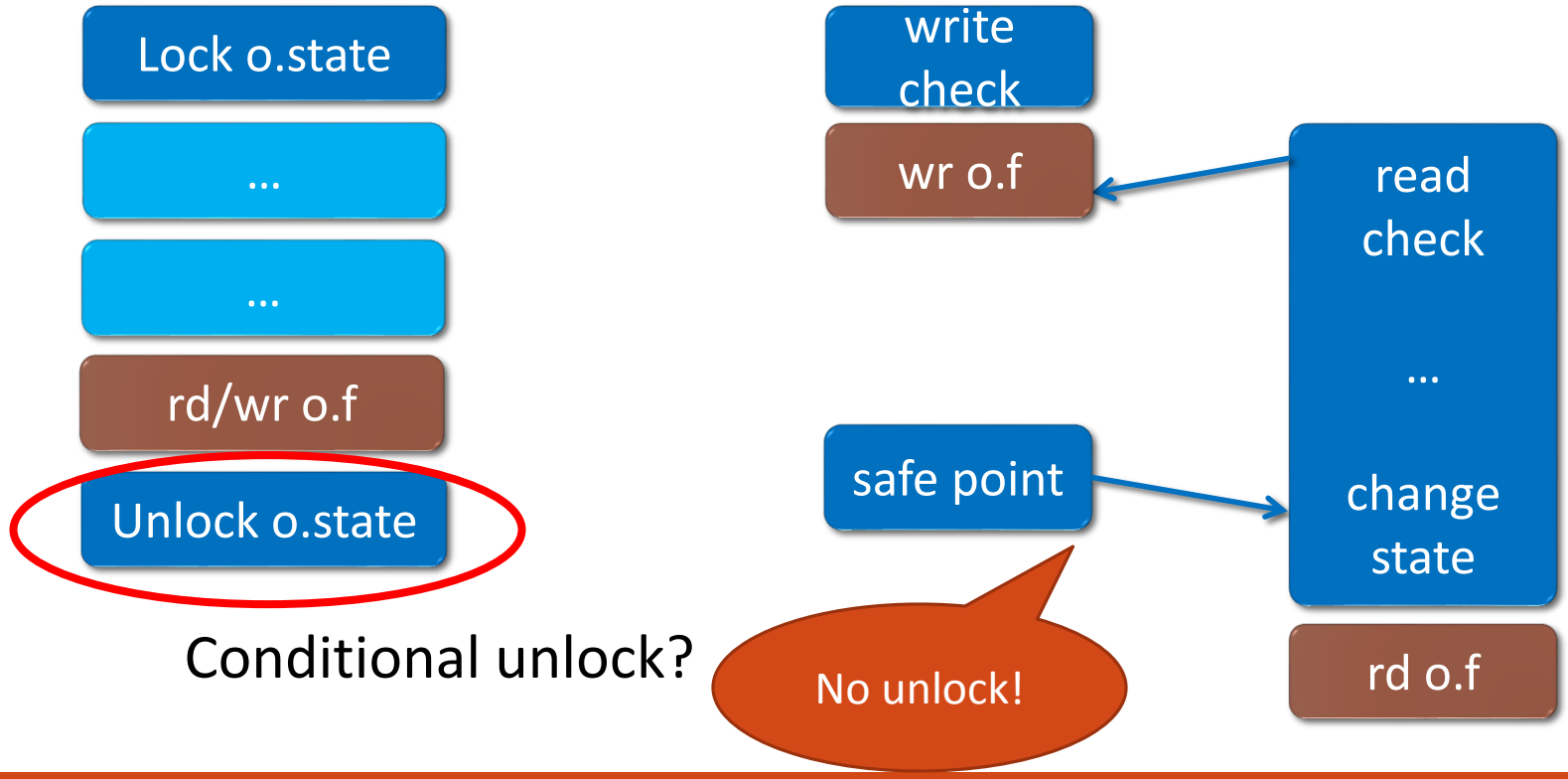


Pessimistic-Optimistic Mismatch (#1)

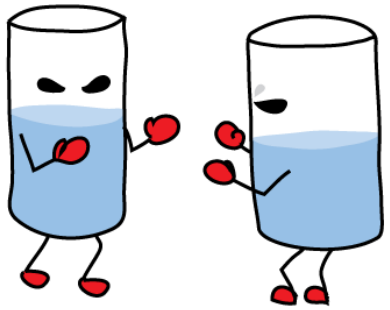


Pessimistic Tracking

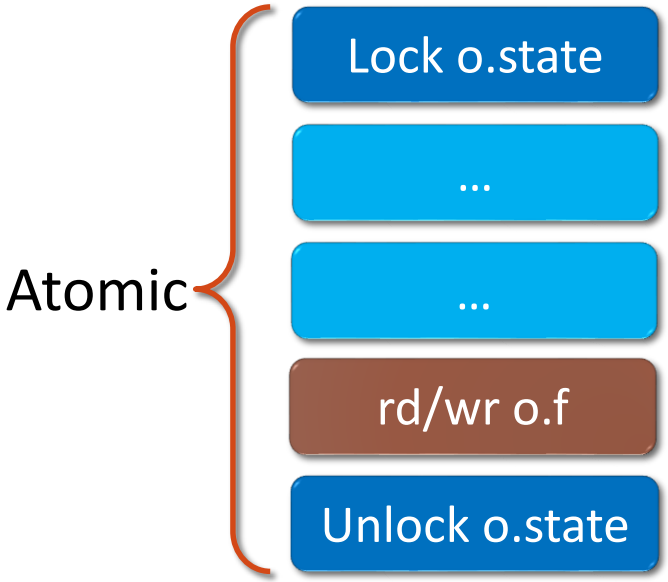
Optimistic Tracking



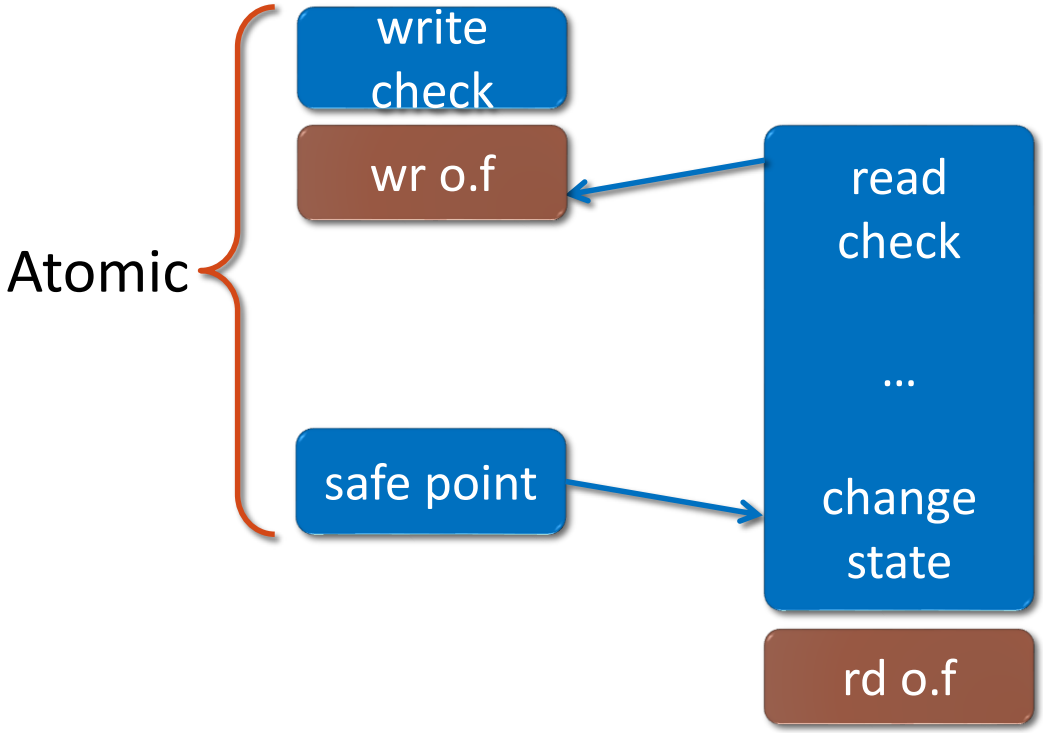
Pessimistic-Optimistic Mismatch (#2)



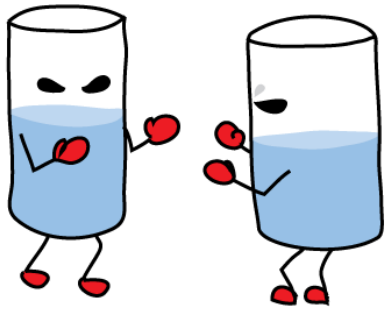
Pessimistic Tracking



Optimistic Tracking

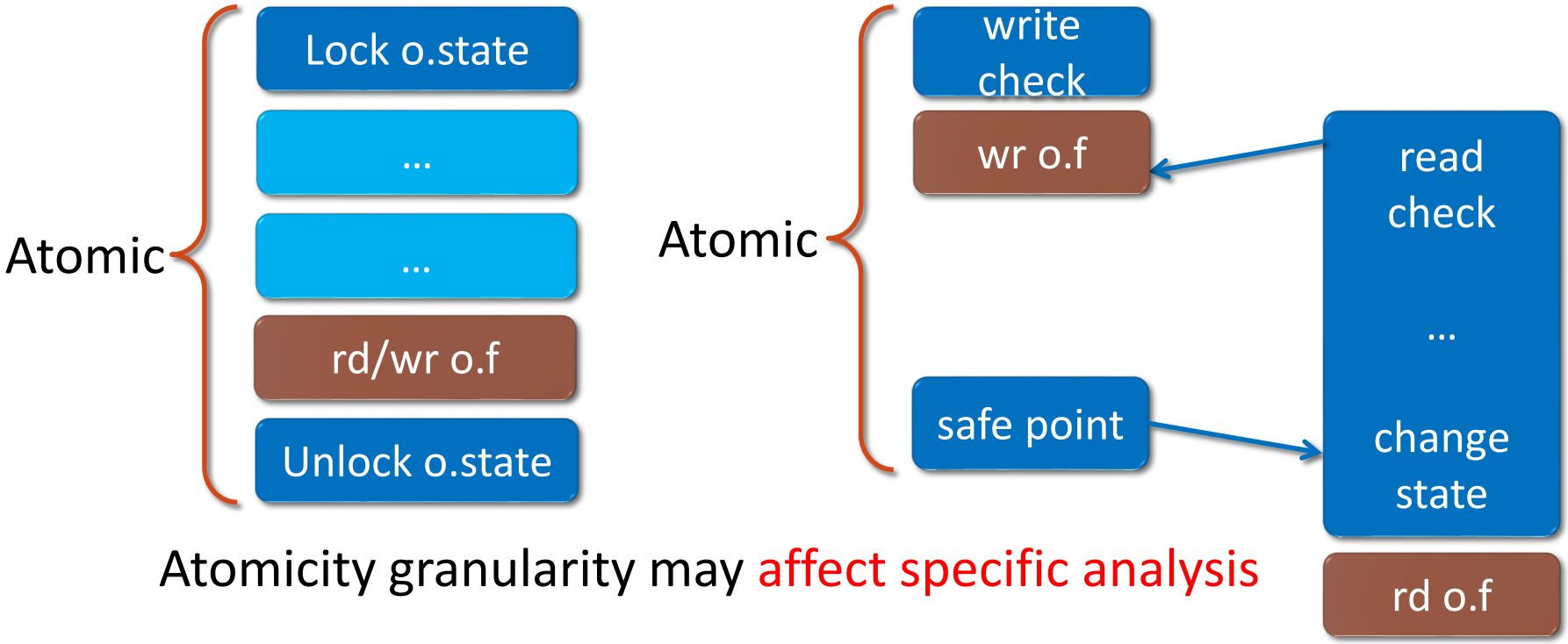


Pessimistic-Optimistic Mismatch (#2)



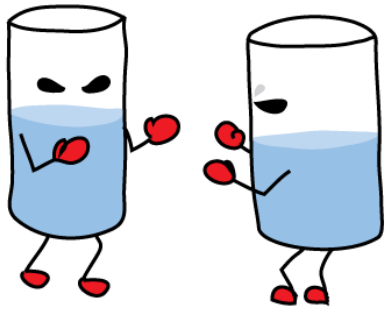
Pessimistic Tracking

Optimistic Tracking

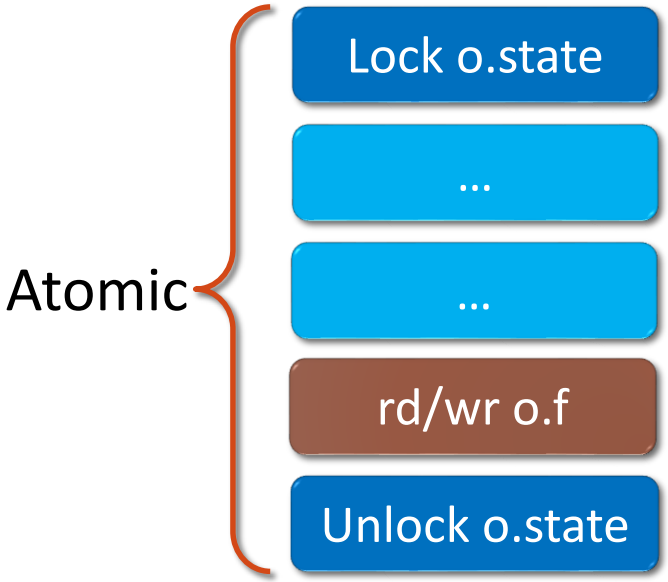


Atomicity granularity may affect specific analysis

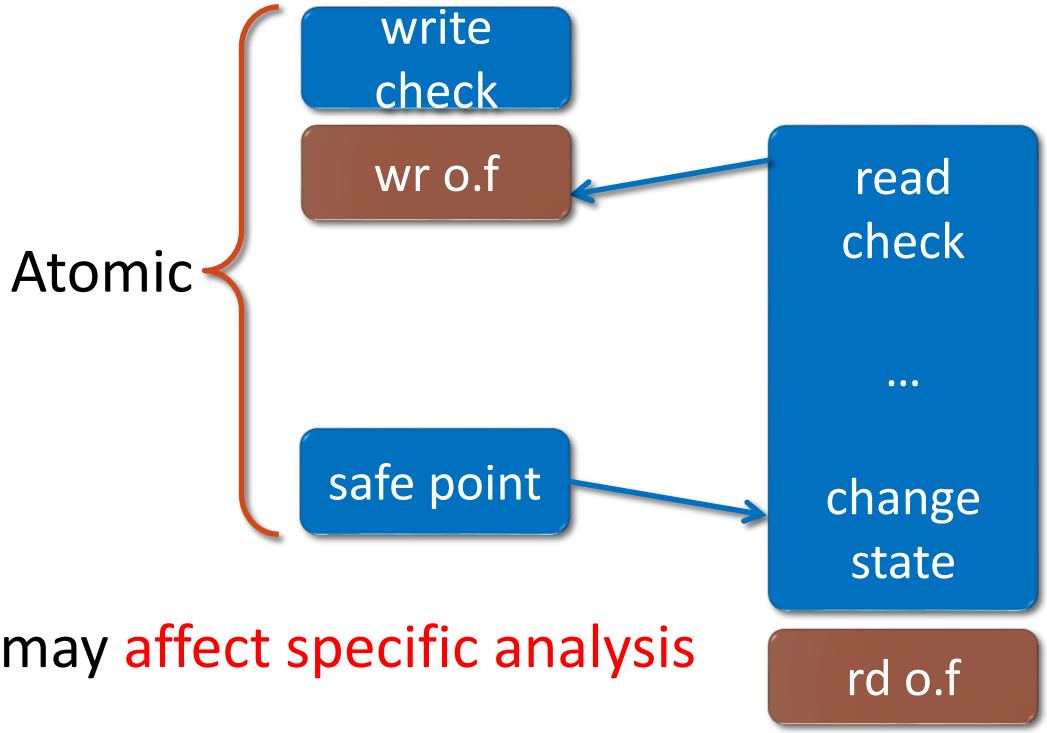
Pessimistic-Optimistic Mismatch (#2)



Pessimistic Tracking



Optimistic Tracking



Atomicity granularity may affect specific analysis

~~Conditional unlock~~

Key Insights

- Coarsening atomicity granularity for pessimistic tracking

Key Insights

- Coarsening atomicity granularity for pessimistic tracking

- Program synchronization may hint at cross-thread dependences

Addressing Pessimistic-Optimistic Mismatch

Defer unlocking of pessimistic state

- Till program synchronization release operation (PSRO)

Addressing Pessimistic-Optimistic Mismatch

Defer unlocking of pessimistic state

- Till program synchronization release operation (PSRO)
- Reader-writer locking

Addressing Pessimistic-Optimistic Mismatch

Defer unlocking of pessimistic state

- Till program synchronization release operation (PSRO)
- Reader-writer locking
- Fall back to coordination on contention when locking a state

Deferred Unlocking Example 1

synchronized (m) {

Lock o.state

wr o.f

…

Unlock all states

}

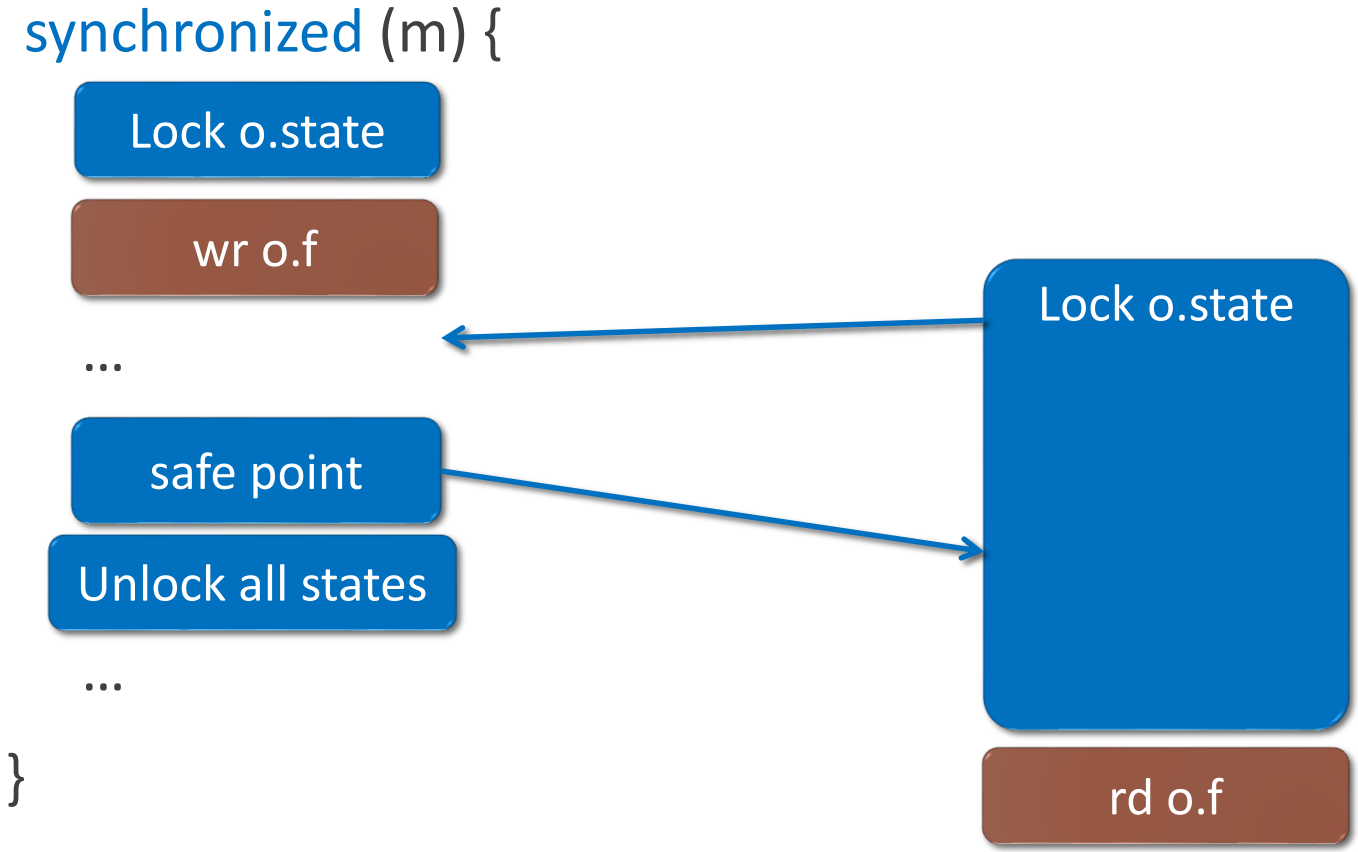
synchronized (m) {

Lock o.state

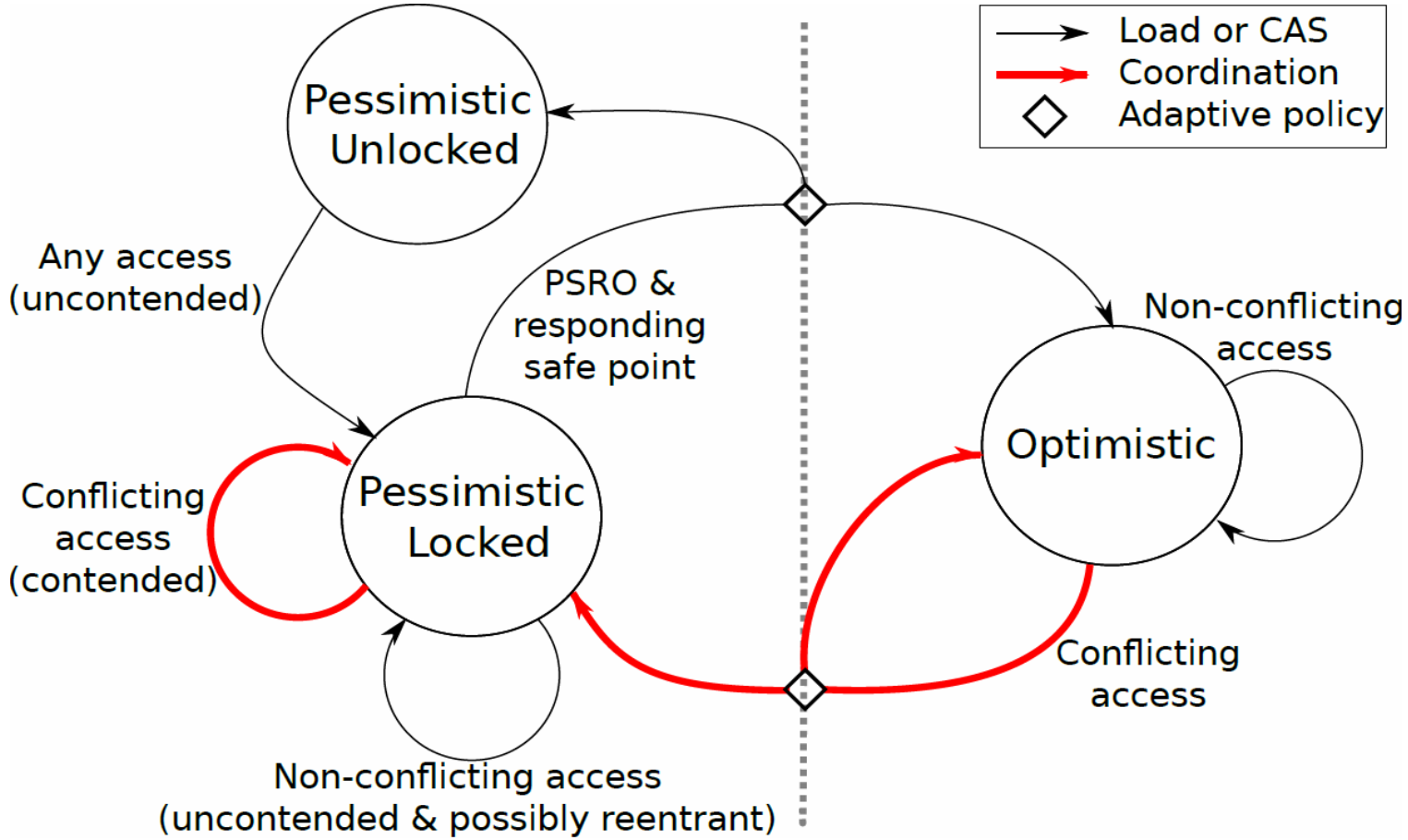
rd o.f

…

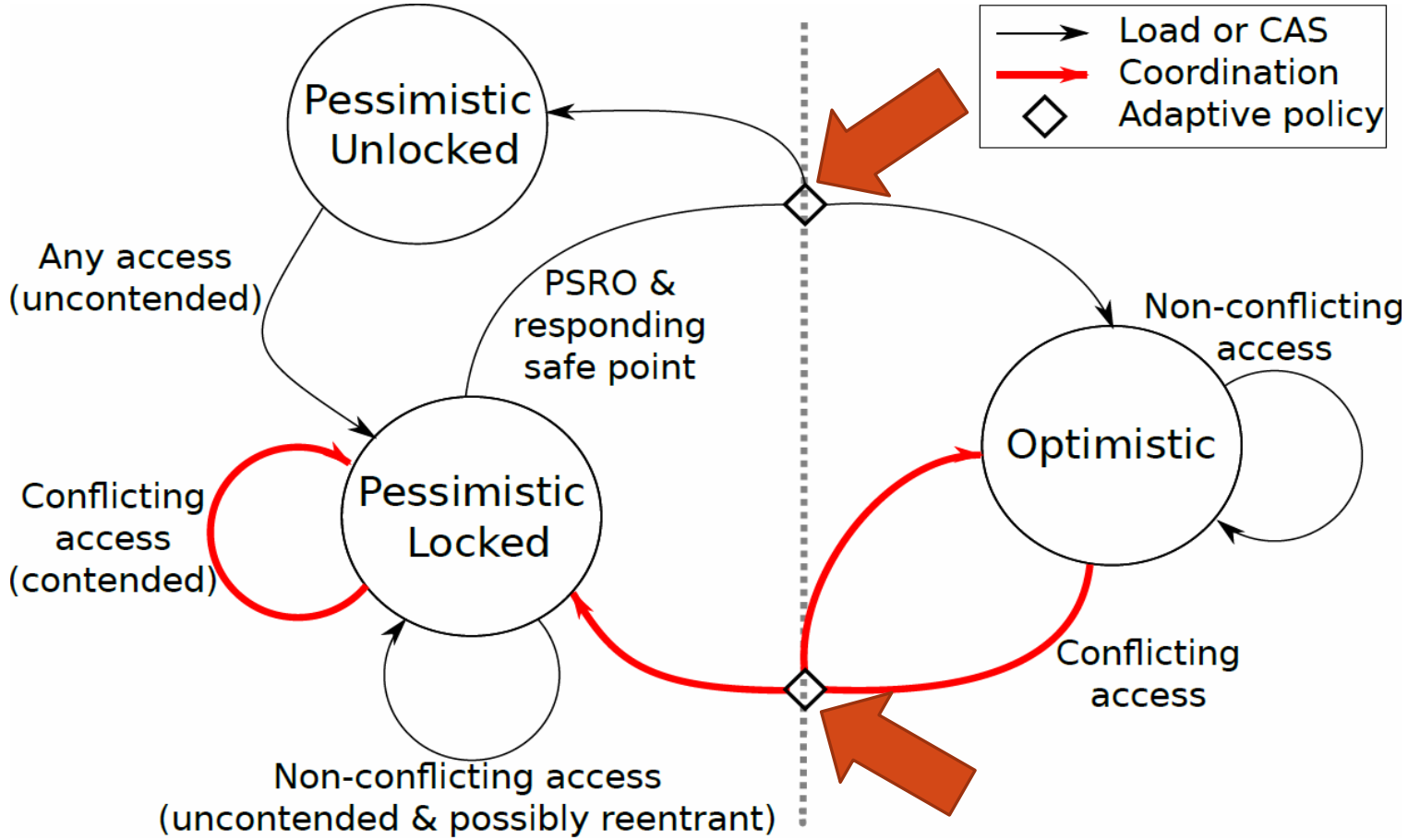
Deferred Unlocking Example 2



Hybrid State Model



Hybrid State Model



Outline

- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Hybrid State Model
Deferred Unlocking
Adaptive Policy

Adaptive Policy

Decide when to transition between pessimistic and optimistic states

Adaptive Policy

Decide when to transition between pessimistic and optimistic states

- Cost—benefit model

Adaptive Policy

Decide when to transition between pessimistic and optimistic states

- Cost—benefit model
  - Boil down to counting state transitions

Adaptive Policy

Decide when to transition between pessimistic and optimistic states

- Cost—benefit model
  - Boil down to counting state transitions
- Online profiling
  - Per-object
  - Simple yet effective

Application of Hybrid Tracking

Two dynamic analyses

- Hybrid dependence recorder and replayer (detect)
- Hybrid region serializability (RS) enforcer (control)

Application of Hybrid Tracking

Two dynamic analyses

- Hybrid dependence recorder and replayer (detect)
- Hybrid region serializability (RS) enforcer (control)

Deferred unlocking helps overcome key challenges!

Outline

- Dynamic Analyses and Cross-thread Dependences
- Pessimistic Tracking
- Optimistic Tracking
- Our approach
  - Hybrid Tracking
- Evaluation

Implementation

Jikes RVM 3.1.3

Implementation

Jikes RVM 3.1.3

Pessimistic tracking, optimistic tracking

- [Octet, Bond et al. OOPSLA'13]

Optimistic recorder and replayer

- [Replay, Bond et al. PPPJ'15]

Optimistic RS enforcer

- [EnfoRSer, Sengupta et al, ASPLOS'15]

Implementation

Jikes RVM 3.1.3

Pessimistic tracking, optimistic tracking

- [Octet, Bond et al. OOPSLA'13]

Optimistic recorder and replayer

- [Replay, Bond et al. PPPJ'15]

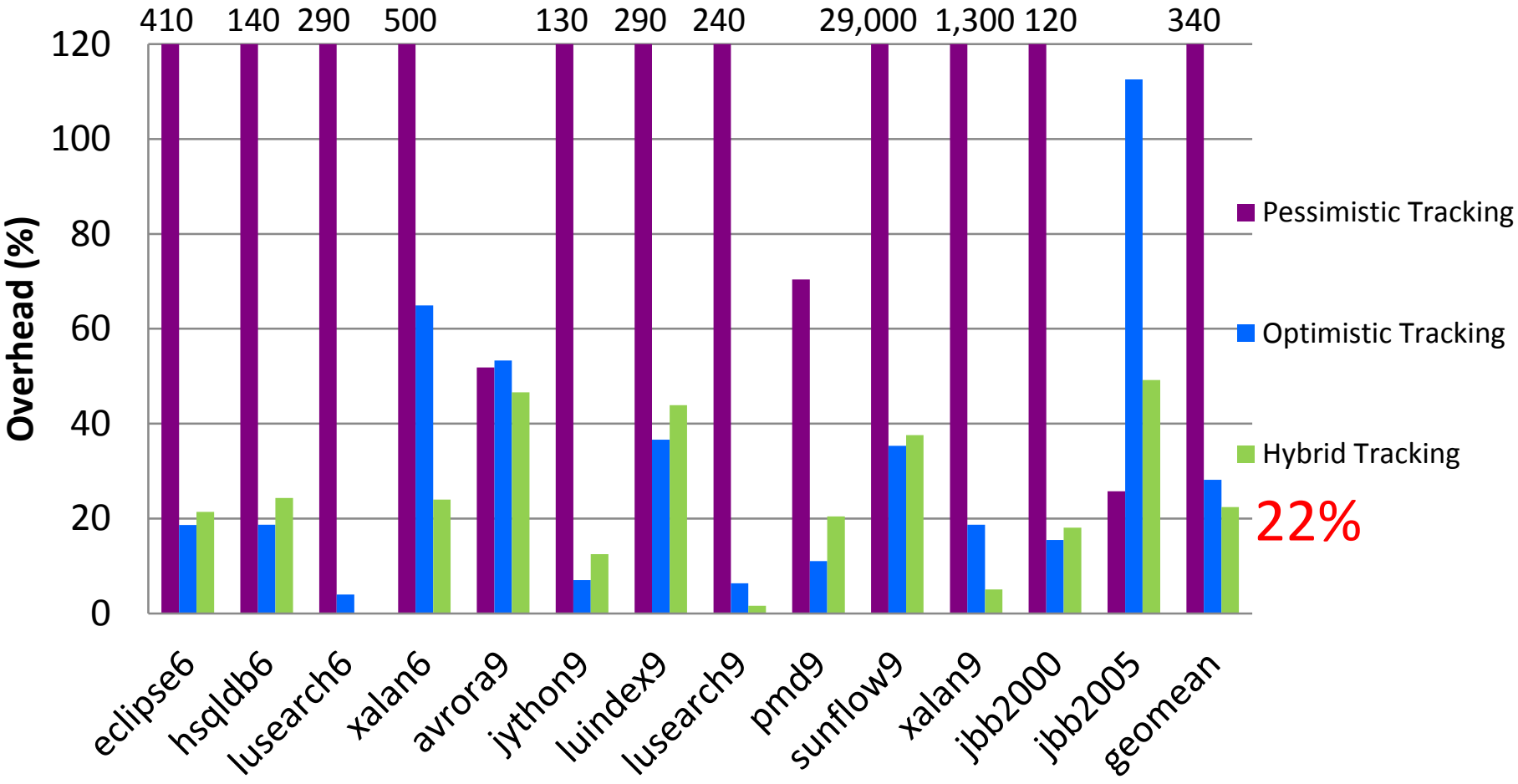
Optimistic RS enforcer

- [EnfoRSer, Sengupta et al, ASPLOS'15]

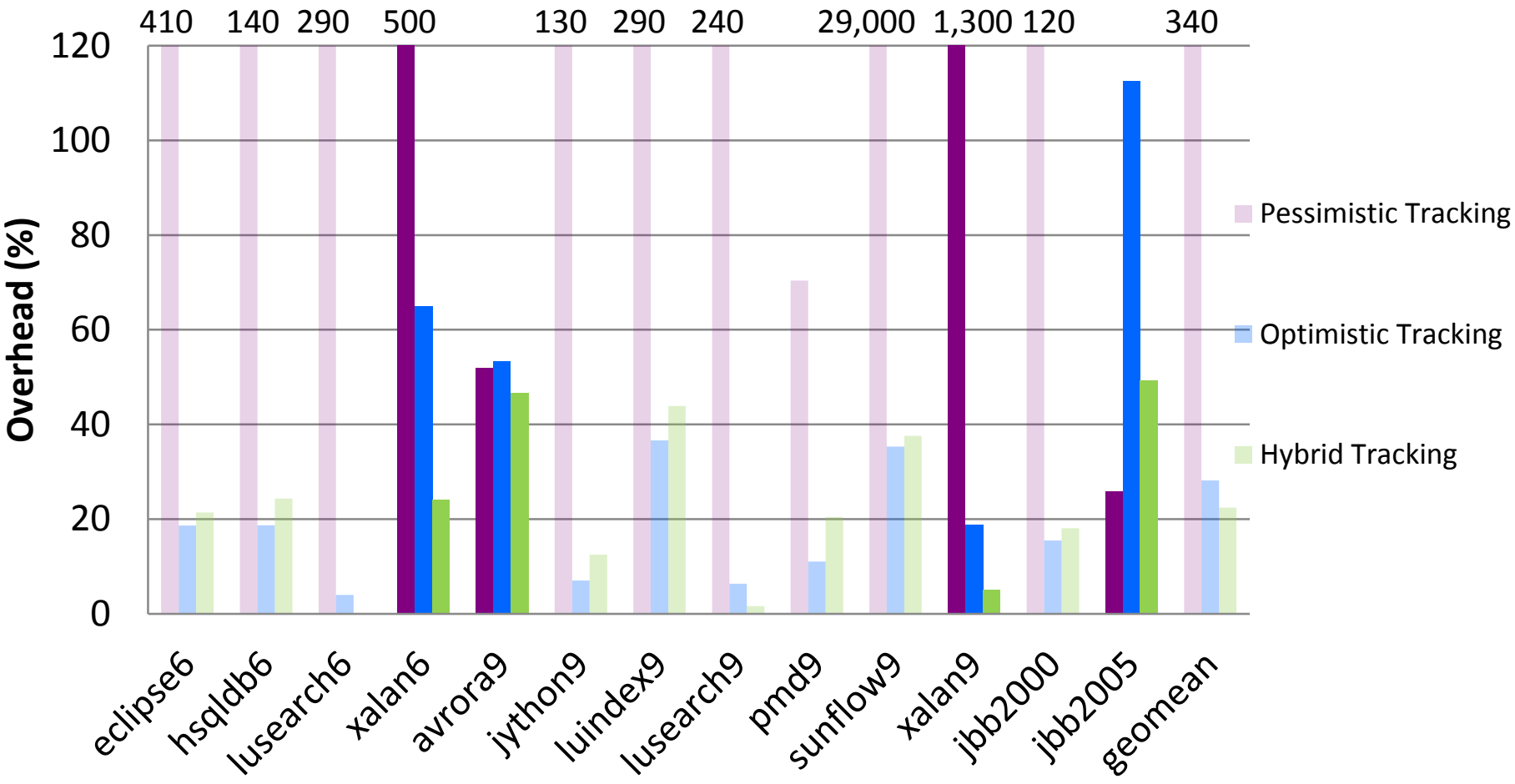
Hybrid tracking, hybrid recorder and replayer, hybrid RS enforcer

- publicly available

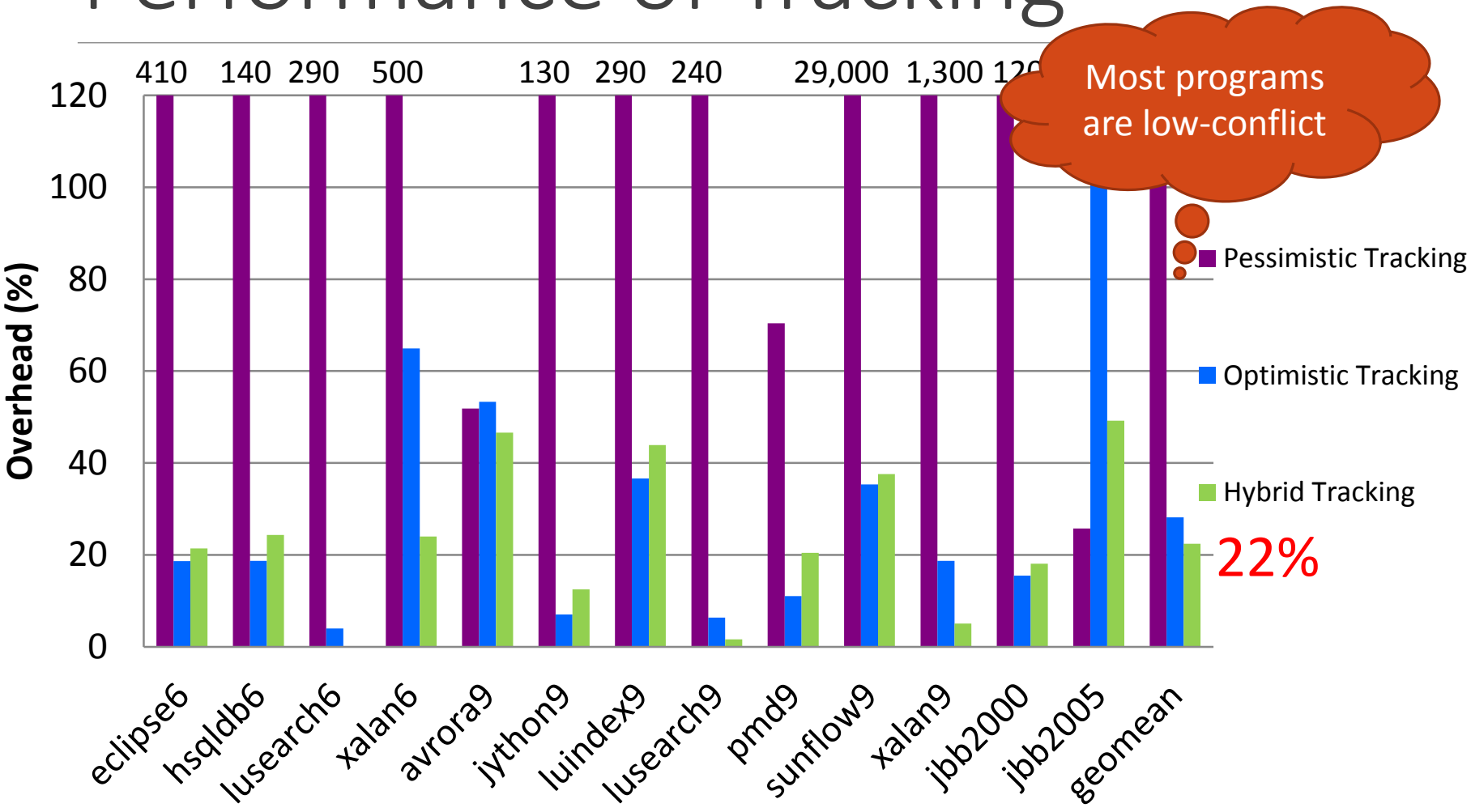
Performance of Tracking



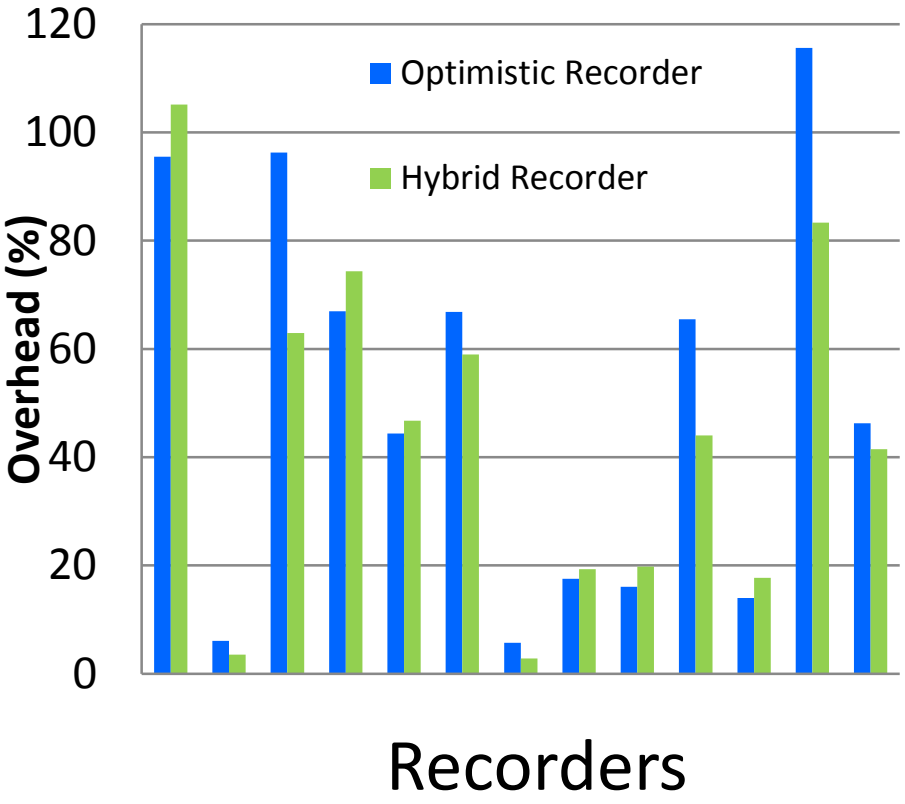
Performance of Tracking

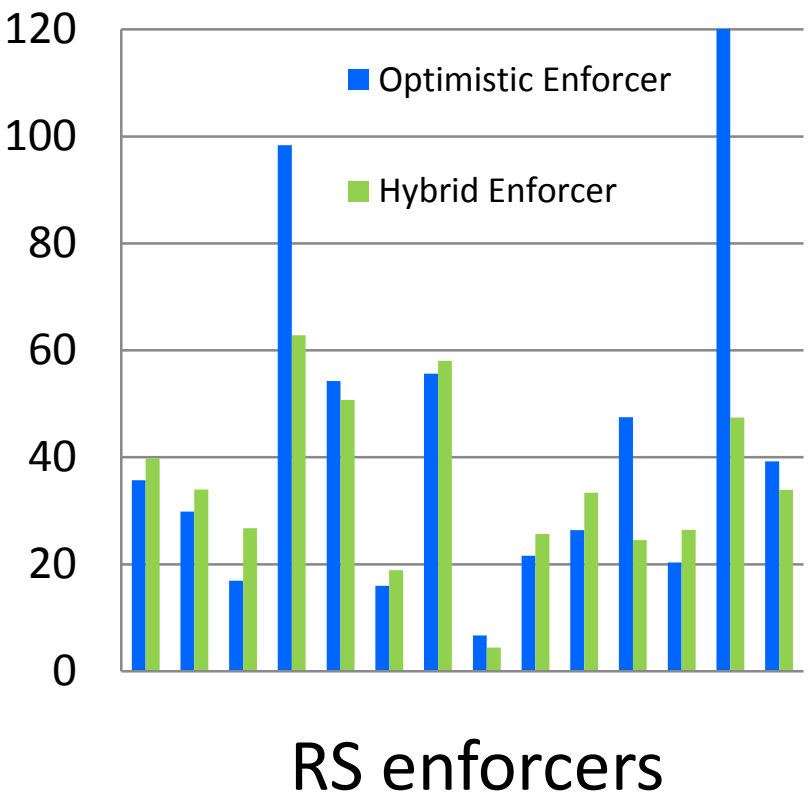


Performance of Tracking

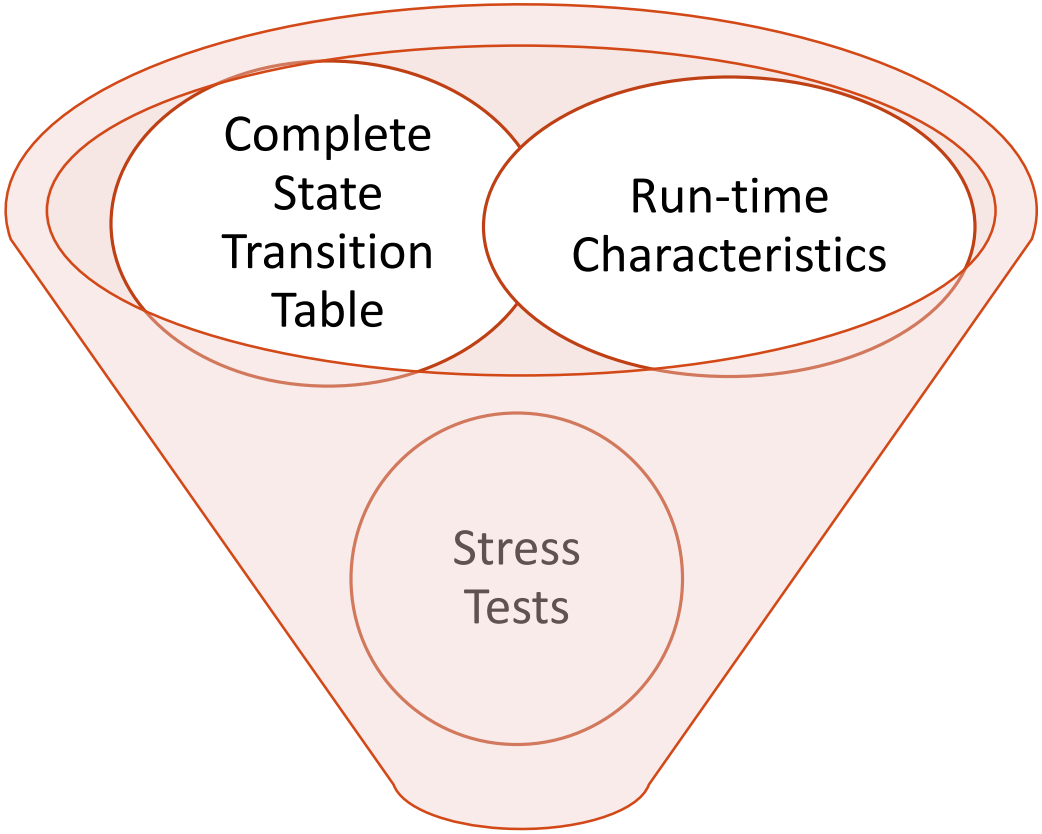


Performance of Recorders and RS enforcers





Additional Materials



Please check the paper

Related work

Analyses that use pessimistic tracking

- [FastTrack, Flanagan & Freund, PLDI'09]
- [Velodrome, Flanagan et al., PLDI'08]
- [Chimera, Lee et al., PLDI'12]
- [Lightweight Transactions, Harris & Fraser, OOPSLA'03]
- [DMP, Devietti et al., ASPLOS'09]

Analyses that use optimistic tracking

- [Shasta, Scales et al. ASPLOS'96]
- [Object Race Detection, von Praun & Gross, OOPSLA'01]
- [DoubleChecker, Biswas et al. PLDI'14]
- [LarkTM, Zhang et al, PPoPP'15]

Adaptive Mechanisms

- [Adaptive Locks, Usui et al. PACT'09]
- [Strong Atomicity TM, Abadi et al. PPoPP'09]
- [Adaptive Lock Elision, Dice et al, SPAA'14]
- [Concurrency Control, Ziv et al, PLDI'15]

Contributions

Hybrid tracking combines pessimistic tracking and optimistic tracking effectively and efficiently

Hybrid tracking achieves better overall performance

- Never significantly degrades performance
- Sometimes improves performance substantially
- Suitable for workload of diverse communication patterns