Predicting Serializability Violations: SMT-based Search vs. DPOR-based Search

Arnab Sinha, Sharad Malik, Princeton University

Resilient Systems, Task 5.5.1

GSRC Annual Symposium

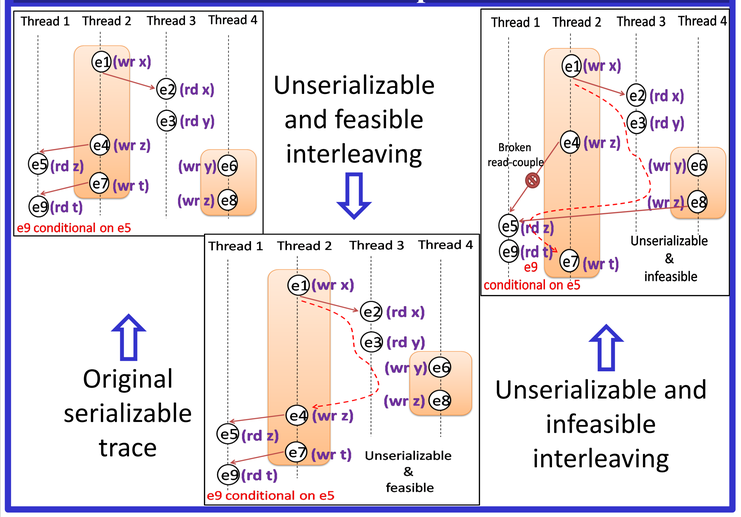
Nov 16 2011

PubID: **2646**

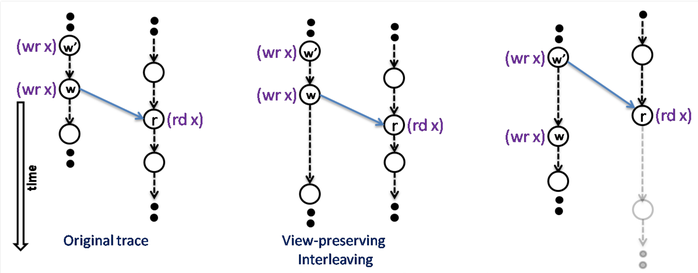
Motivation

- ❑ > 69% of concurrency bugs are atomicity violations.
  - ❑ detected by checking for conflict-serializability.
- ❑ Checking alternate interleavings of a given trace (Predictive Analysis) is a compromise between formal verification and monitoring.
- ❑ For predictive analysis, we can efficiently bound the interleaving search space using a trace slice (Trace Atomicity Segment – TAS) in practice (MEMOCODE'11).
- ❑ This work compares explicit (Dynamic Partial Order Reduction (DPOR)) and implicit (SMT) techniques for search-space exploration.

An Example



Almost View Preserving (AVP) Interleavings

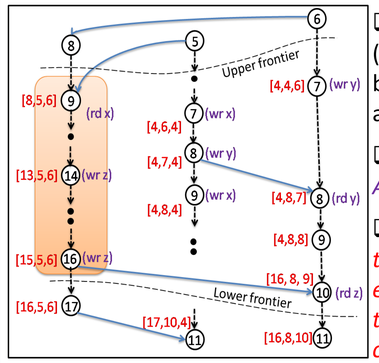


An event *v* is *skipped* in an interleaving ρ, if v_r must happen before *v* s.t. v_r is the read in a broken read-couple in ρ.

Acknowledgements

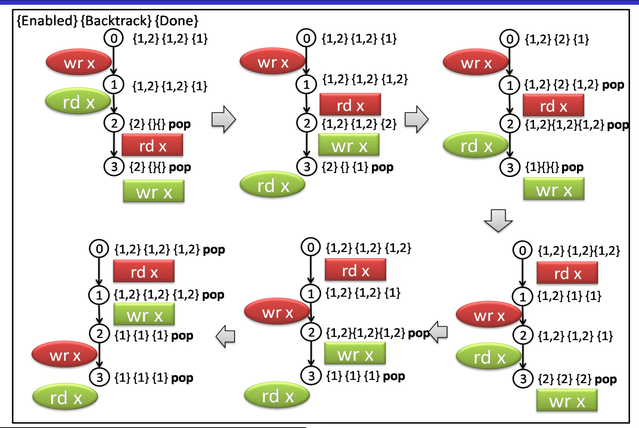
The authors acknowledge the support of the Gigascale Systems Research Center. They also thank their collaborators: Dr. Chao Wang and Dr. Aarti Gupta.

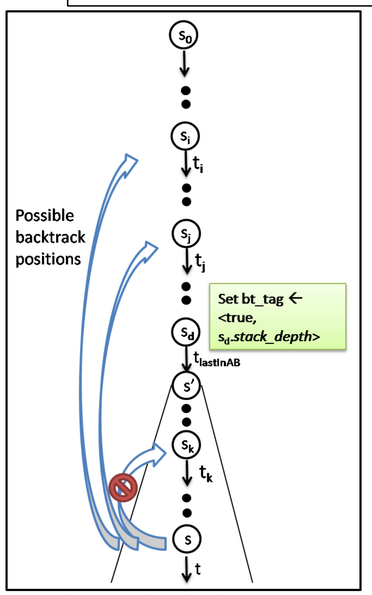
Trace Atomicity Segment (TAS)



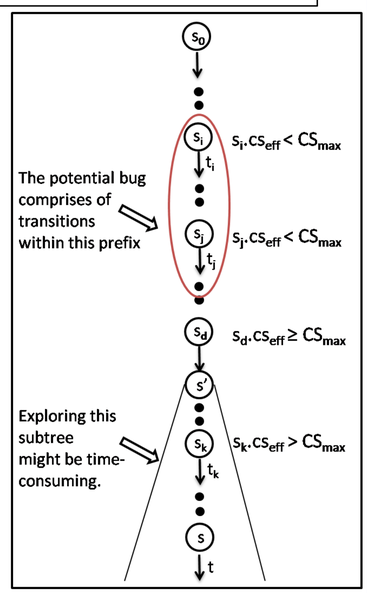
- ❑ Events above (below) the upper (lower) frontier must happen before (after) the atomic block in all AVP interleavings.
- ❑ Frontiers bound the *Trace Atomicity Segment* (TAS).
- ❑ Key Result: *No error-path for the atomic block can contain events outside the TAS. This limits the set of interleavings considered.*

Pruning in Dynamic Partial Order Reduction (DPOR)



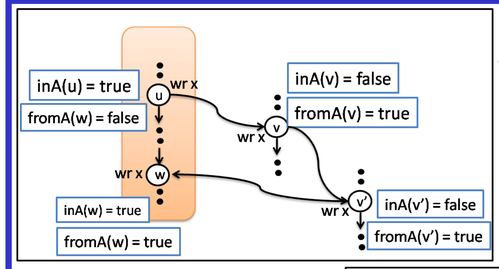


A marker set for the last transition ($t_{lastInAB}$) from the atomic region in the prefix helps pruning the interleavings.



Tuning the parameter CS_{max} can help in carrying out localized search.

SMT Formulation



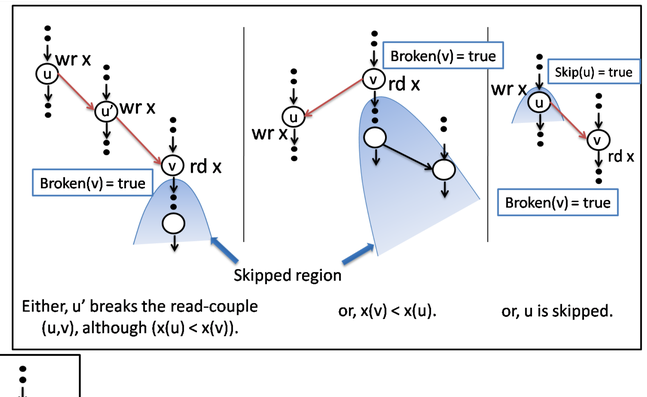
❑ $\Phi_{VP} = \bigvee_{\forall u \in V(\mathcal{A})} fromA(u)$ denotes the violation path, where,

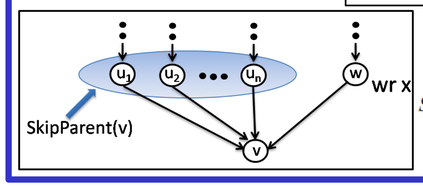
$$fromA(v) = \bigvee_{\forall (u,v) \in EP\_Set} edgeFromA(u,v)$$

$$edgeFromA(u,v) = (\neg inA(v) \ \wedge \ (x(u) < x(v))) \quad \text{if } u \in V(\mathcal{A})$$

$$edgeFromA(u,v) = (fromA(u) \ \wedge \ (x(u) < x(v))) \quad \text{if } u \notin V(\mathcal{A})$$

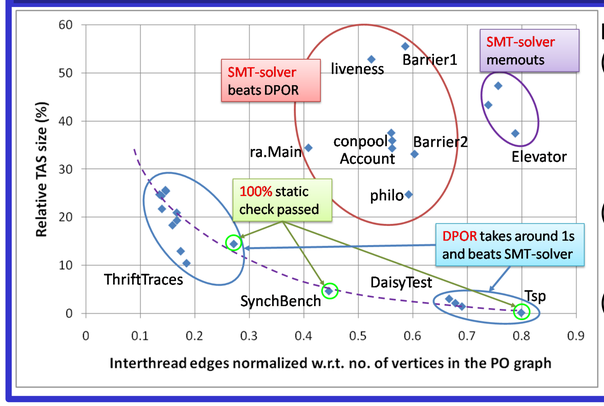
❑ AVP interleavings are captured with predicates *Broken* and *Skip*.





$$Skip(v) = \begin{cases} \text{false} & \text{if } SkipParent(v)=\{\} \\ \bigvee_{u \in SkipParent(v)} Skip(u) \ \vee \ Broken(u) & \\ \text{otherwise.} & \end{cases}$$

Results



Limited to our data-set ,

(1) If coupling strength and relative TAS size indirectly proportional, TAS+DPOR is more effective since less interleavings to explore, and,

(2) Little above the curve TAS+SMT is more effective (more interleavings).

(3) Beyond a certain threshold, SMT-solvers give up.

Contributions

Several major contributions in predictive analysis for atomicity checking:

- ❑ Tuned DPOR technique to predict serializability violations among AVP interleavings.
- ❑ Proposed SMT-based encoding for predicting serializability violations with any number of threads and variables for the first time.
- ❑ Identified some characteristics of instances that can be used to select between them.