Predictive Analysis for Detecting Serializability Errors Through Trace Segmentation

Arnab Sinha, Sharad Malik, Princeton University
Resilient Systems, Task 5.5.1

GSRC Annual Symposium
Sep 28-Oct 1, 2010

Motivation
- More than 69% of concurrency bugs are atomicity violations.
- Detected by checking for conflict-serializability.
- Checking a single trace (trace monitoring) is efficient but limited.
- Checking all possible traces through formal verification is intractable.
- Checking alternate interleavings of a given trace (Predictive Analysis) is a compromise between formal verification and monitoring.
- Still intractable due to the large number of interleavings. Need to prune number of interleavings considered.
- Need to ensure feasible interleavings to avoid false positives.

An Error Path
An error path $P(p')$ in a feasible interleaving $p'$ is a simple path of alternating (RD/WR) vertices and conflict edges in $p'$ s.t.
- It originates and terminates in $A$.
- It visits at least one vertex outside $A$.
- Atomicity violation for $A \not\equiv$ Error path $P(p')$.

The Methodology
Partial-order Graph:
- Vertices: events (read/write, synch. events, atomic begin & end)
- Edges: program order, read-after-writes, wait-notify, fork-first event, terminate-join.

An Example
Read-couples: $(e_1,e_2)$ & $(e_2,e_3)$.
Original program trace might be bug-free.

Almost View Preserving (AVP) Interleavings
An event $v$ is skipped in an interleaving $p$, if $v$, must happen before $v$ s.t. $v$, is the read in a broken read-couple in $p$.

Results
- The static check is very effective. Avg. pass rate is 78.2%.
- In SynchBench, 100% of atomic blocks pass.
- Less than 1% of checks timeout.
- Avg. TAS size is only 26.2% (best case Tsp 0.2%).
- TAS needed for non-terminating/ streaming applications.
- TAS + DPOR works well for both tightly & loosely coupled threads.

Related Work
- Semantic Analysis (performance bottleneck!)
  - partial interpretation of transactions [Rosu et al. TR UIUC]
  - Trace and source code analysis [Wang et al. TACAS’10]
- Complexity Study (bogus errors reported too!)
  - Predicted errors are not guaranteed to show up. [Farzan and Madhusudan CAV’09, TACAS’09]
- Runtime monitoring (performance bottleneck!)
  - Reports real bugs; performance bottleneck [Rosu et al. CAV’07, Sinha et al. IPDPS’10]

Contributions
Several major contributions in predictive analysis for atomicity checking:
- Almost View Preserving (AVP) interleavings as a useful set of feasible interleavings.
- Sufficiency of Trace Atomicity Segment (TAS) for considering all error paths in all AVP interleavings.
- Practical utility of the static check (enabled by the TAS).
- Validation through detailed experiments.