동시성 커버리지 기반 멀티쓰레드 프로그램 테스트 자동생성

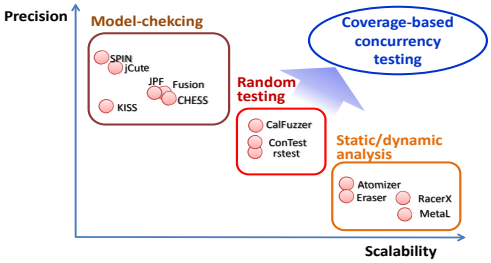
홍신, 김문주 | SWTV, KAIST

Overview

- Generating a test to achieve **high concurrency coverage fast** is effective and efficient to detect concurrency errors in multithreaded programs
- The **CUVE** testing technique generates thread schedules that achieve high coverage fast

Motivation and Idea

- Existing concurrency testing/analysis techniques suffer from false alarm problems and/or state-explosion problems



- *Concurrency coverage metrics* can provide useful measure & guidance for thread schedule gen.
  - Ex. Sync-pair coverage metric

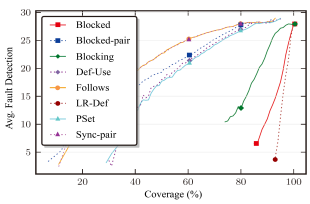
```
10:foo() {
11: lock(m);
12: unlock(m);
13: lock(m);
14: unlock(m);
15:}
```

```
20:bar() {
21: lock(m);
22: unlock(m);
23: lock(m);
24: unlock(m);
25:}
```

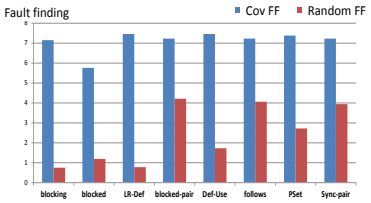
Total SP test requirements:
(11, 13), (11, 21), (11, 23), (13, 21), (13, 23), (21, 11), (21, 13), (21, 23), (23, 11), (23, 13)

Are Concurrency Coverage Metrics Really Effective for Concurrency Testing?

- Empirically evaluate eight concurrency coverage metrics and their relations to fault findings
- Q1: Does coverage impact fault finding?
  A1: *Yes*. The metrics estimate properly estimate testing effectiveness.
- Q2: Is a test controlled to achieve high coverage more effective than a random test?
  A2: *Yes*. Generating a test toward high coverage can detect more fault than random testing



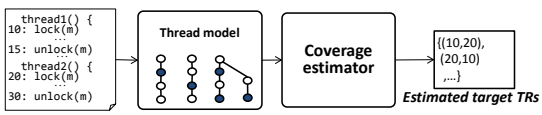
Cov vs. Fault finding, Vector object



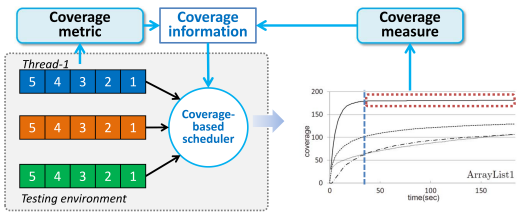
Cov-test vs. Random-test, ArrayList object

How to Make Tests To Achieve High Concurrency Coverage?

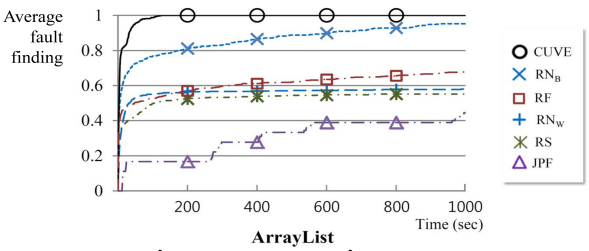
- Control execution orders of threads to achieve test req. from concurrency coverage metrics
- CUVE technique
  - Phase I: estimates achievable test requirements



  - Phase II: generates thread schedules by
    - monitor running thread status and measure cov. suspend/resume threads to cover uncovered test requirements



- Experiment result
  - Compare fault finding ability of CUVE with random testing techniques, a model checker, and other heuristic-based testing techniques
  - CUVE shows **higher fault finding** than the existing testing techniques
  - CUVE is **faster to achieve high fault finding** than the existing testing techniques



On-going and Future Work

- Case study of real-world multithreaded program
  - A project with Samsung Electronics to apply CUVE to find concurrency errors in the field
- Regression testing of multithreaded programs
  - Concurrency coverage can capture new or modified thread interaction by code change
  - Use CUVE to selectively generate test exec at code changes by targeting impacted test req.