Log-Powered Test Scenario Generation for Distributed Systems

Ivan Beschastnikh
Yuriy Brun
Michael D. Ernst
Arvind Krishnamurthy
Thomas E. Anderson

http://synoptic.googlecode.com

University of Washington
**Synoptic (our prior work)**

A tool that mines FSM models from logs

Input

Output

http://synoptic.googlecode.com
How Synoptic works

Parses the log with user-defined reg. expressions

Invariants

Mines temporal log invariants

e.g.,

"lock() is always followed by unlock()"

"open() always precedes read()"

Input

Output

Finds a compact model satisfying mined invariants

propose

tx-abort

tx-commit

open()

invariants

Finds a compact

www.synoptic.googlecode.com

"lock() is always followed by unlock()"

"open() always precedes read()"
### Generating new test cases

The FSM is generative: Use it to predict new executions

---

**Input**

```
src : 0, dst : 2, timestamp : 6, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 0, dst : 2, timestamp : 22, type : ack
src : 2, dst : 1, timestamp : 23, type : tx_commit
src : 1, dst : 2, timestamp : 23, type : ack
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 2, dst : 0, timestamp : 16, type : prepare
src : 1, dst : 2, timestamp : 7, type : ack
```

**Output**

```
commit
abort
tax-commit
tax-abort
```

---

**Synoptic**

A tool that mines FSM models from logs

---

```
exec: propose
exec: abort
exec: commit
exec: tx-commit
exec: tx-abort
```
Generating new test cases

The FSM is generative: Use it to predict new executions

But... why??

Synoptic

A tool that mines FSM models from logs

http://synoptic.googlecode.com
Generating new test cases

The FSM is generative: Use it to predict new executions

But... why??

A tool that mines FSM models from logs

new executions ≈ new tests
Generating new test cases

- Generate an FSM from executions log
  - By running Synoptic
- Generate a plausible execution (sequence of strings)
  - By traversing the generated FSM
- Map the plausible execution to a path in the program
  - Leverage SherLog
- Generate inputs to induce the path
  - Leverage ideas from Concolic testing
- Run!

http://synoptic.googlecode.com
Generating new test cases

1) Contradictory constraints (no solution): improve model

- Generate a plausible execution (sequence of strings)
  - By traversing the generated FSM

- Map the plausible execution to a path in the program
  - Leverage SherLog

- Generate inputs to induce the path
  - Leverage ideas from Concolic testing

- Run!
Generating new test cases

1) Contradictory constraints (no solution): improve model

2) Insolvable: try another execution

- Generate an FSM from executions log
  - By running Synoptic
- Generate a plausible execution (sequence of strings)
  - By traversing the generated FSM
- Map the plausible execution to a path in the program
  - Leverage SherLog
  - Yuan et al. ASPLOS 2010
- Generate inputs to induce the path
  - Leverage ideas from Concolic testing
- Run!
Generating new **test cases**

1) Contradictory constraints (no solution): improve model

2) Insolvable: try another execution

3) Runs ok: passing test-case

- Generate an FSM from executions log
- By running Synoptic
- Generate a plausible execution (sequence of strings)
- By traversing the generated FSM
- Map the plausible execution to a path in the program
- Leverage SherLog
- Generate inputs to induce the path
- Leverage ideas from Concolic testing
- Run!

http://synoptic.googlecode.com
Generating new test cases

1) Contradictory constraints (no solution): improve model

2) Insolvable: try another execution

3) Runs ok: passing test-case

4) Crash: failing test-case

Run!
Wait, what about distributed systems?
Extending to distributed systems

- How do you replicate what I’ve talked about for concurrent systems?

- What is different:
  - Logged executions are DAGs, not strings
  - The FSM model is no longer applicable (need a different model type)
  - Symbolic execution is much more difficult
  - Can solve separately for each process
  - Enrich constraints to include remote events

http://synoptic.googlecode.com
Extending to distributed systems

- How do you replicate what I’ve talked about for concurrent systems?

- What is different:
  - Logged executions are DAGs, not strings
  - The FSM model is no longer applicable (need a different model type)
  - Symbolic execution is much more difficult
  - Can solve separately for each process
  - Enrich constraints to include remote events

many unsolved problems = ripe for research