

# Log-Powered Test Scenario Generation for Distributed Systems

---

Ivan Beschastnikh

Yuriy Brun

Michael D. Ernst

Arvind Krishnamurthy

Thomas E. Anderson



University of Washington





# How Synoptic works

```
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
```

Parses the log with user-defined reg. expressions

```
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
```

Input

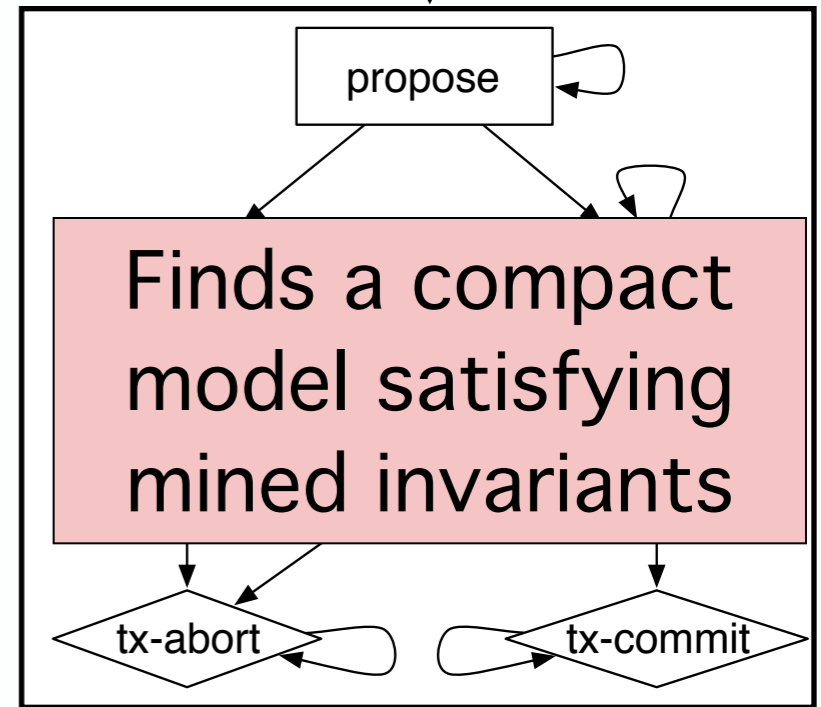
Invariants

Mines temporal log invariants

e.g.,

“lock() is always followed by unlock()”

“open() always precedes read()”



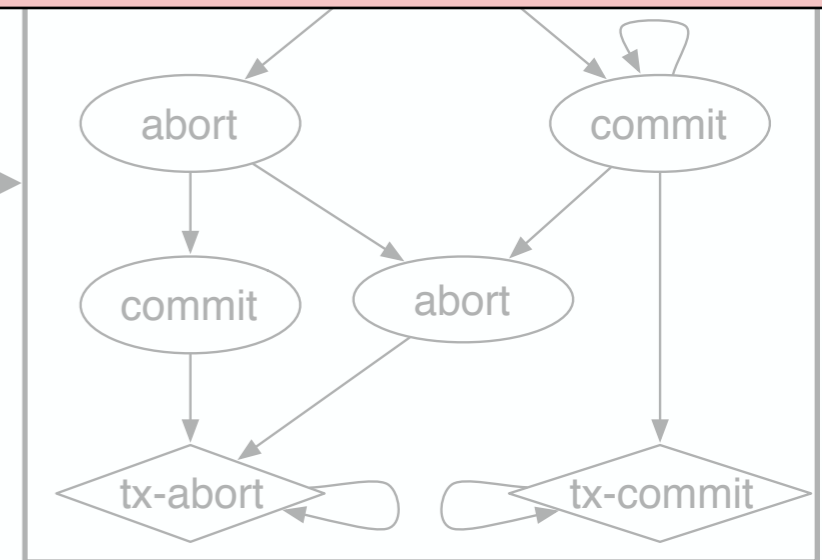
Output

# Generating new test cases

The FSM is generative:  
Use it to predict new executions

Synoptic

A tool that mines FSM models from logs



Output

```
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 2, dst : 0, timestamp : 0, type : prepare
src : 2, dst : 1, timestamp : 1, type : prepare
src : 0, dst : 2, timestamp : 2, type : commit
src : 1, dst : 2, timestamp : 3, type : commit
src : 2, dst : 0, timestamp : 4, type : tx_commit
src : 2, dst : 1, timestamp : 5, type : tx_commit
src : 0, dst : 2, timestamp : 6, type : ack
src : 1, dst : 2, timestamp : 7, type : ack
src : 2, dst : 0, timestamp : 8, type : prepare
src : 2, dst : 1, timestamp : 9, type : prepare
src : 0, dst : 2, timestamp : 10, type : commit
src : 1, dst : 2, timestamp : 11, type : commit
src : 2, dst : 0, timestamp : 12, type : tx_commit
src : 2, dst : 1, timestamp : 13, type : tx_commit
src : 0, dst : 2, timestamp : 14, type : ack
src : 1, dst : 2, timestamp : 15, type : ack
src : 2, dst : 0, timestamp : 16, type : prepare
src : 2, dst : 1, timestamp : 17, type : prepare
src : 0, dst : 2, timestamp : 18, type : commit
src : 1, dst : 2, timestamp : 19, type : commit
src : 2, dst : 0, timestamp : 20, type : tx_commit
src : 2, dst : 1, timestamp : 21, type : tx_commit
src : 0, dst : 2, timestamp : 22, type : ack
src : 1, dst : 2, timestamp : 23, type : ack
src : 1, dst : 2, timestamp : 19, type : commit
```

Input



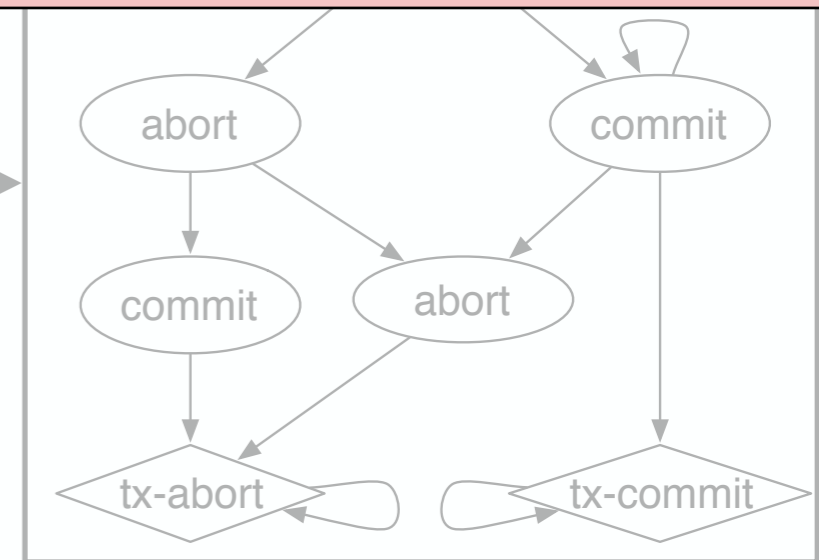
# Generating new **test cases**

**The FSM is generative:  
Use it to predict new  
executions**

**But...  
why??**

Synoptic

A tool that mines FSM  
models from logs



Output

**new executions  $\approx$  new tests**

Input

# 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 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**
- 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**

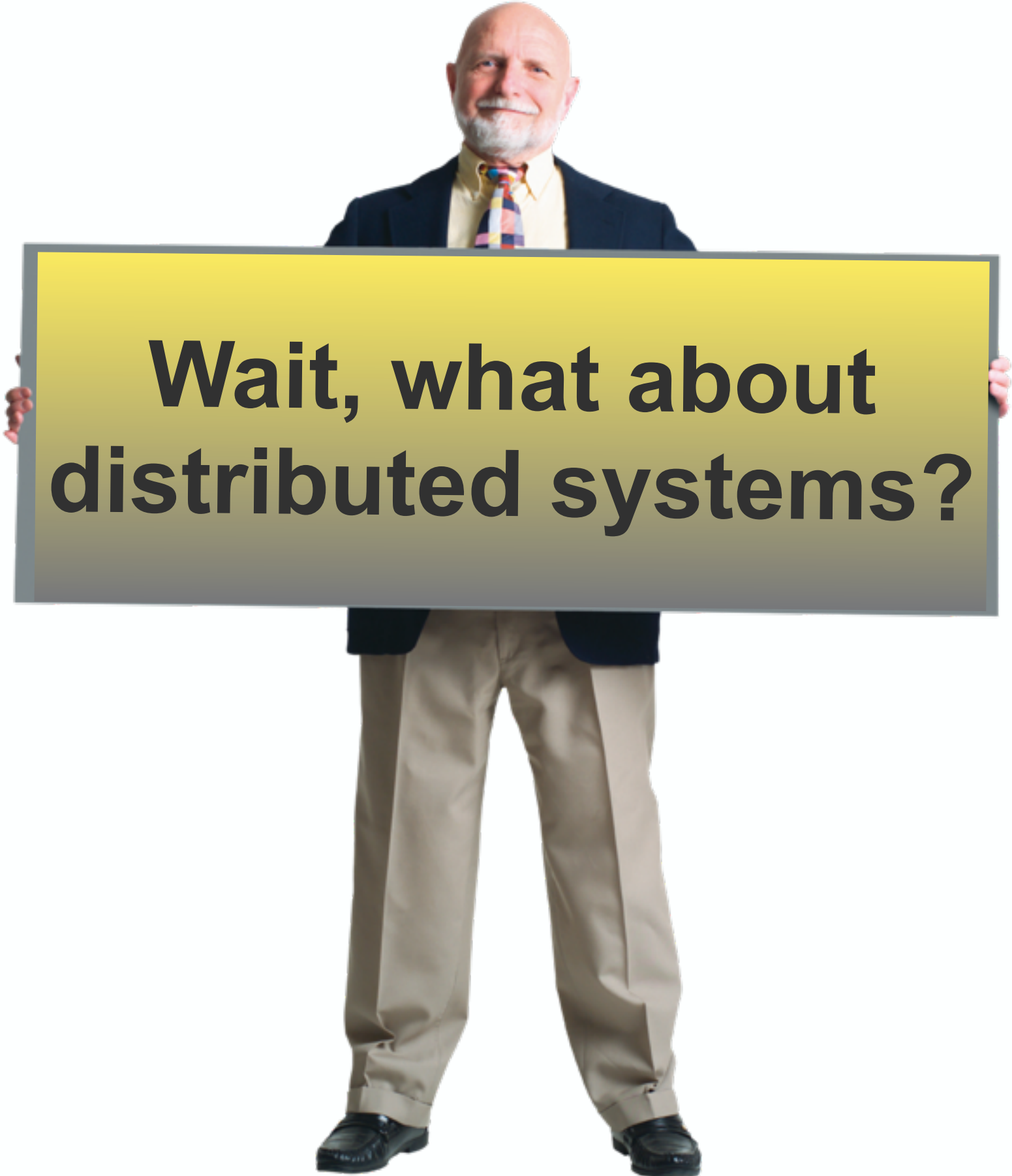
- **1) Contradictory constraints (no solution): improve model**
- Generate a plausible execution (sequence of strings)
- **2) Insolvable: try another execution**
- Model
- Leverage SherLog Yuan et al. ASPLOS 2010
- Generate inputs to induce the path
  - Leverage ideas from Concolic testing
- Run!

# Generating new **test cases**

- **1) Contradictory constraints (no solution): improve model**
- Generate a plausible execution (sequence of strings)
- **2) Insolvable: try another execution**
- Model
- **3) Runs ok: passing test-case**
- Generate inputs to induce the path
  - Leverage ideas from Concolic testing
- Run!

# Generating new **test cases**

- **1) Contradictory constraints (no solution): improve model**
- Generate a plausible execution (sequence of strings)
- **2) Insolvable: try another execution**
- Model
- **3) Runs ok: passing test-case**
- Generate inputs to induce the path
- **4) Crash: failing test-case**
- Run!

A full-body photograph of a man with a white beard and balding head, wearing a dark suit jacket, a light-colored shirt, and a colorful tie. He is holding a large, rectangular sign in front of his chest. The sign has a yellow-to-white gradient background and contains the text "Wait, what about distributed systems?".

**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

# 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**