Inferring data invariants in distributed systems
https://bitbucket.org/bestchai/dinv

Debugging distributed systems is difficult and error-prone:

**Dinv:** tool to infer **Distributed system Invariants**

- Works on systems written in Go
- Requires annotations to know when to log state
- Determines distributed state with slicing analyses
- Instruments code to record partial order and concrete state values during execution
- Infers likely invariants over recorded values at consistent cuts using Daikon

**Dinv instrumentation overview**

1. Developer annotates code
   ```
   i := 1
   sum := 0
   product := 1
   for i <= n {
     sum := sum + i
     product := product * i
     i := i + 1
   }
   ```

2. Dinv adds instrumentation
   ```
   // dump
   ```

3. Developer runs the system
   ```
   product := product * i
   ```

4. System generates logs of node states
   ```
   A.log
   B.log
   C.log
   ```

5. Dinv associates node states using consistent cuts
   ```
   A.cnt <= B.cnt
   B.leader ∈ {A, C}
   ```

6. Daikon infers invariants from values at matching cuts

**Research questions**

- What is the annotation effort?
- What is the performance overhead?
- Are Daikon relations sufficient?
- Is grouping program points by consistent cuts too fine-grained?

**Applications**

- Regression/bug detection
- Characterizing test suite deficiency
- System comprehension

Stewart Grant, Sam Creed, Ivan Beschastnikh
University of British Columbia