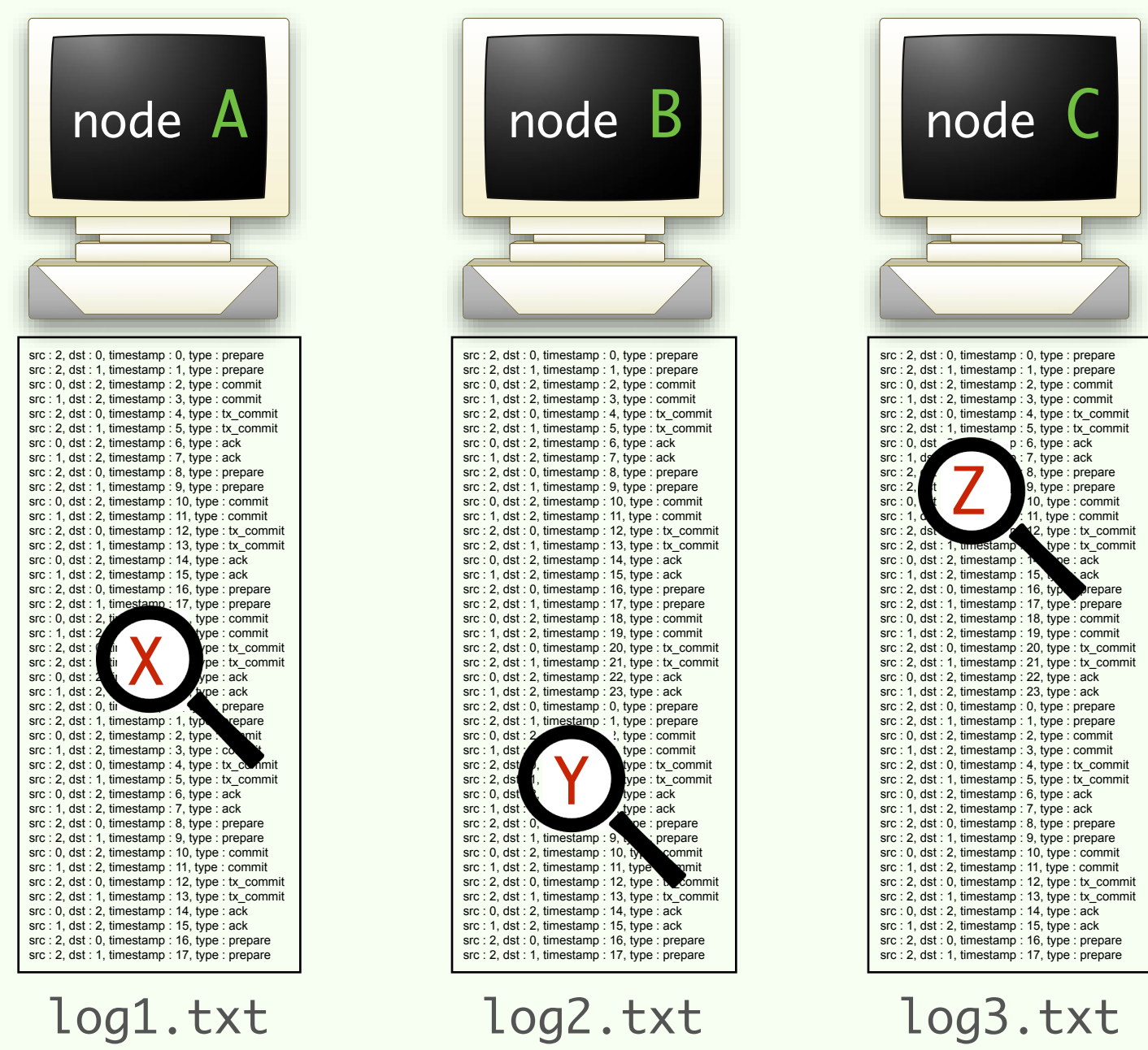


Inferring data invariants in distributed systems

<https://bitbucket.org/bestchai/dinv>

Debugging distributed systems is difficult and error-prone:

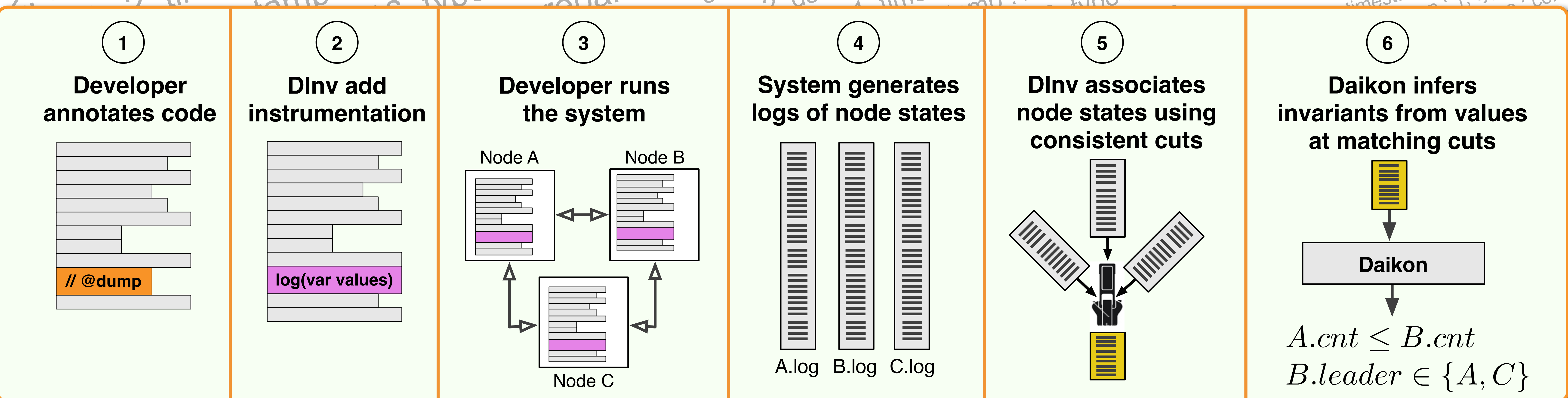


Requires knowing the distributed state and its properties across nodes:

- What variables at program point **X** are influenced by node **B**?
- How does the distributed state at **Y** change?
- Which node's send causes the state at node **A** to change?
- Do values at program points **X** and **Z** ever differ?

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 recv(n)
2 i:= 1
3 sum := 0
4 product := 1
5 for i <= n {
6   sum := sum + 1
7   product := product * i
8   i := i + 1
9 }
10 send(sum)
11 // @ dump
12 send (product)
    
```

Developer adds **dump** annotations at key program points

```

1 recv(n)
2 i:= 1
3
4 product := 1
5 for i <= n {
6
7   product := product * i
8   i := i + 1
9 }
10
11 // @ dump
12 send (product)
    
```

Backward slice: code affecting the sent **product** variable

```

1 recv(n)
2 i:= 1
3
4 product := 1
5 for i <= n {
6
7   product := product * i
8   i := i + 1
9 }
10
11 // @ dump
12 send (product)
    
```

Variables appearing in the slice: **i, n, product**

```

1 recv(n)
2 i:= 1
3 sum := 0
4 product := 1
5 for i <= n {
6   sum := sum + 1
7   product := product * i
8   i := i + 1
9 }
10 send(sum)
11 point = {[i,n,product],vclock}
12 Log(point)
13 send (product)
    
```

Injected code to log **product**-affecting vars

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