## 1. Motivation

Soft errors occurring during a program execution may silently corrupt the output.

- Hardware-level fault tolerance techniques are costly.
- Application-level techniques are essential.
- Memory systems largely protected using ECC techniques.
- Goal: Detect/recover from soft errors in CPU elements.

### 2. Fault Detection by Tracking Predicate Transitions

**Predicate Transition Diagram of Foo() function**

```
int Foo(int x, int y)
{
    if ( x<7  &&  y>10 )
    {
        PP0: y++;    
        PP1: x++;    
        PP2: } else 
        PP3: 
        PP4:   
        PP5: printf("x=%d, y=%d", x , y)
    }
}
```

- **Predicates:** x<7, y>10
- **Program Points:** PP0, PP1, PP2, PP3, PP4, PP5
- **Input Test Vector:** x=6, y=11
- **Predicate State at PP0:** <PP0:TT>
- **Predicate State at PP1:** <PP1:TT>
- **Example Predicate Transition:** <PP0:TT> → <PP1:TT>

### 3. Visualizing Spurious Predicate Transitions

- Predicate Transition Diagram of Foo() function
- Spurious (red-dashed) transitions caused by soft errors
- Based on the idea of predicate abstraction
- Used by Ball to define novel coverage metrics [6]
- Using it to build soft error detectors is a novel direction!

### 4. FUSED Soft-Error Detection Framework

- Automatically synthesizes and inserts detectors.
- Likely invariants are used for soft errors detection.
- Uses profilers to generate likely invariants.
- Previous approach derived likely invariants by tracking program properties [4]
- Our approach derives likely invariants using predicate transitions.

### 5. Preliminary Experimental Results

- **FUSED** is evaluated using SuperLU Library [1][2][3]
- Up to 90% of soft errors are detected.
- Detectors only inserted into top-level LU factorization routine.
- Average execution overhead of 15.7% due to the detectors.

### 6. Conclusions & Future Work

- Introduction to a new soft-error detection framework FUSED which automatically synthesizes and inserts program-level detectors.
- Preliminary evaluation of FUSED framework using SuperLU library shows promise.
- In future, optimize detector placement to further reduce overhead.

### 7. Acknowledgements

- Supported in part by NSF Award CCF 1255776 and SRC contract 2013-TJ-2426.
- Pedro Diniz, Prabhakar Kudva, Shuvendu Lahiri, Karthik Pattabiraman, and Greg Bronevetsky.

### 8. Closely Related References


