RT-DFI: Optimizing Data-Flow Integrity for Real-Time Systems[1]

Nicolas Bellec Guillaume Hiet Simon Rokicki
Frederic Tronel Isabelle Puaut

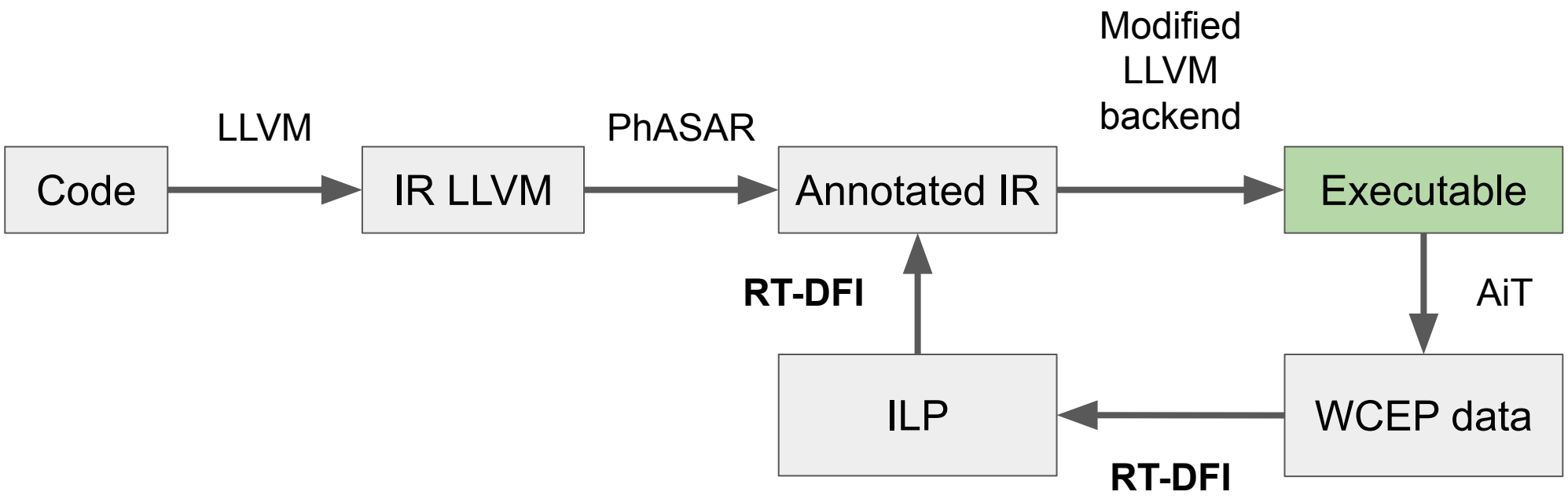
UMR IRISA

Context

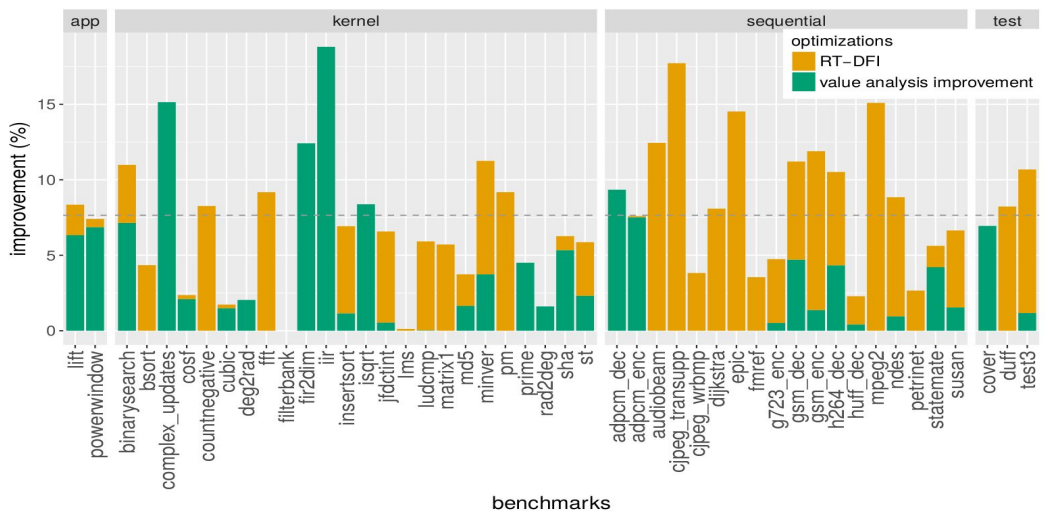
- Real-Time Systems
- Data-Flow Hijacking attacks
  (ex: validating with a wrong password)

Existing Protection: Data-Flow Integrity

Optimization: Focus on the *Worst-Case Execution Path*



Results



- Mean improvement : **7.6%** (compared with no optimization)
- ILP resolution **< 30s**
- Iteration does not further improve the result

[1]RT-DFI: Optimizing Data-Flow Integrity for Real-Time Systems, *Bellec et al*, ECRTS'22

LSChain: Detecting redundancies in Data-Flow Integrity

Nicolas Bellec Guillaume Hiet Simon Rokicki
Frederic Tronel Isabelle Puaut

UMR IRISA

Problem: Redundancy in Data-Flow Integrity

```
store r1, 0 (sp)

store r1, 4 (sp)
```

DFI

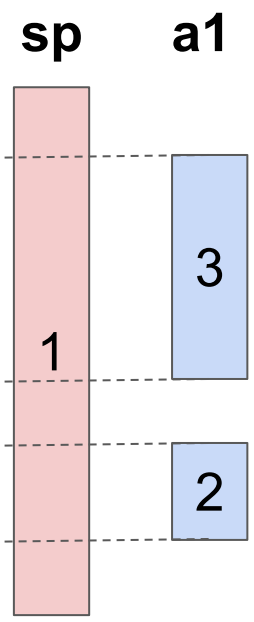
```
dfi_addr t3, 0 (sp)
store_dfi_tag t3, 1
store r1, 0 (sp)

dfi_addr t3, 4 (sp)
store_dfi_tag t3, 2
store r1, 4 (sp)
```

Almost the same
Cost: 3-5 cycles

Can be replaced by an "*add 2*"

Idea: Find chains of loads/store instructions using a same, unmodified register

```
sw s1, 12 (sp)
sw s2, 8 (a1)
lui a0, 261888
sw a0, 24 (sp)
lw a2, 12 (a1)
add a1, a1, a3
sw a2, 4 (a1)
sw s0, 0 (a1)
sw a3, 40 (sp)
```

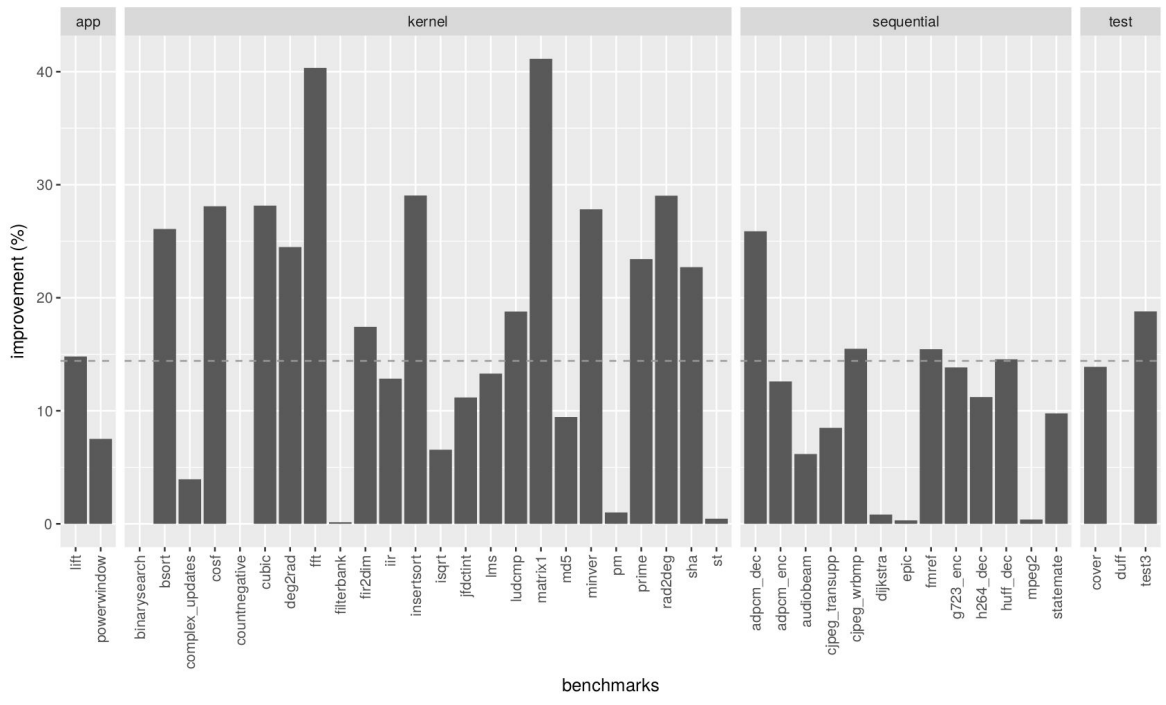


Example with 3 chains

Requirements:

- **Intra basic-block**
- **A register is available along the chain**
- **An offset difference multiple of 4**

Results



- Mean improvement : **14.4 %** (compared with no optimization)
- Use a **greedy heuristic** to select the chain to optimize when not enough free registers

Future Works

- Inter basic-block
- Improved optimization when missing free registers