DITTANY: STRENGTH-BASED DYNAMIC INFORMATION FLOW ANALYSIS TOOL FOR X86 BINARIES

WALID J GHANDOUR AND CLEMENTINE MAURICE

UNIV. LILLE, CNRS, INRIA

Binary Analysis Research (BAR) Workshop 2022

APRIL 24, 2022

INFORMATION FLOW

- Dynamic Information Flow Analysis (DIFA) monitors the flow of information between objects/variables/instructions in a program at runtime
- Flow strength quantifies the amount of info flow between instructions in a program
- Tool support for strength based dynamic dependence analysis and experimental evidence of its effectiveness on the x86 platform.
- Use the tool to introduce:
  - Correlation-based predictors
  - DIFA directed
    - data value predictor
    - indirect branch predictor

PROGRAM SLICING

- Program slice
- Dynamic slice
- Assembly dynamic slice (ADS)

DEPENDENCE STRENGTH

- ***Information flow*** from an instruction *ins1* to an instruction *ins2*
  - if observing the outcome of *ins2* at some point reduces one's uncertainty about the outcome of *ins1* at an earlier point.
- The ***strength of dependences*** measures the amount of information they propagate.

WHAT IS INSTRUMENTATION?

- A technique that inserts extra code into a program to collect runtime information

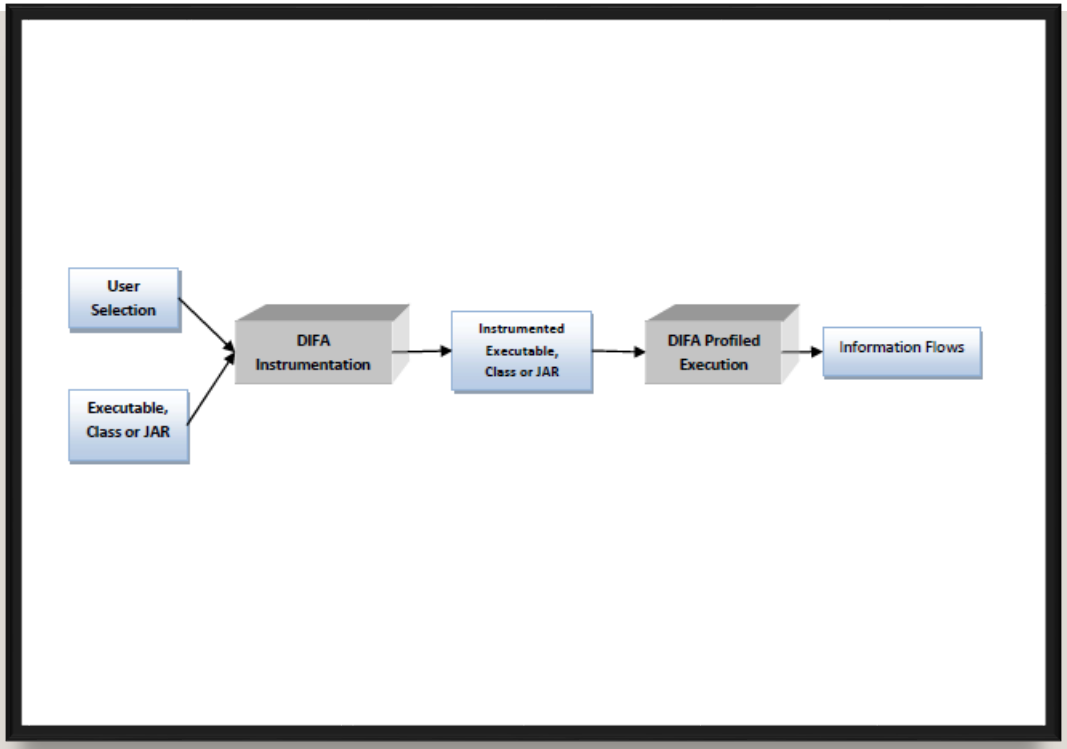
- sub $0xff, %edx
- counter++;
- cmp %esi, %edx
- counter++;
- jle <L1>
- counter++;
- mov $0x1, %edi
- counter++;
- add $0x10, %eax
- counter++;

PIN: DYNAMIC BINARY INSTRUMENTATION TOOL

- Uses dynamic instrumentation
- Programmable Instrumentation
- Multiplatform
- Robust
- Efficient

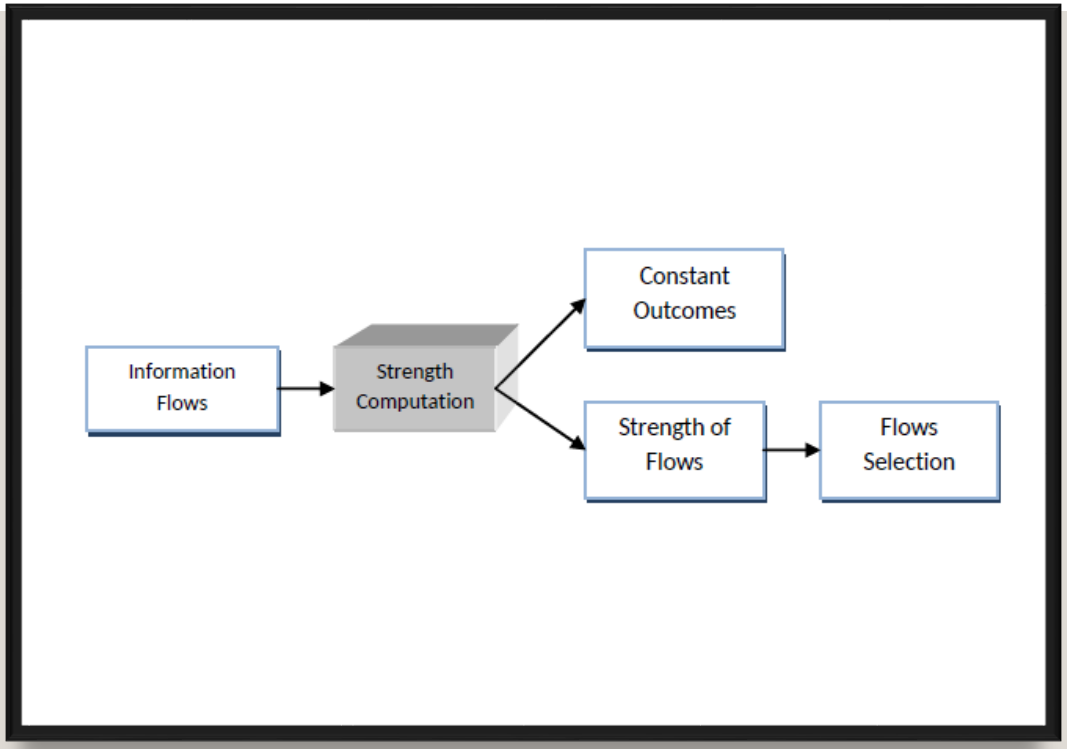
OVERVIEW OF DIFA TOOL OPERATION FOR DYNAMIC INFORMATION FLOW ANALYSIS

- Identifies dynamic dependences
- Records the associated values induced at the sources and targets of dependences

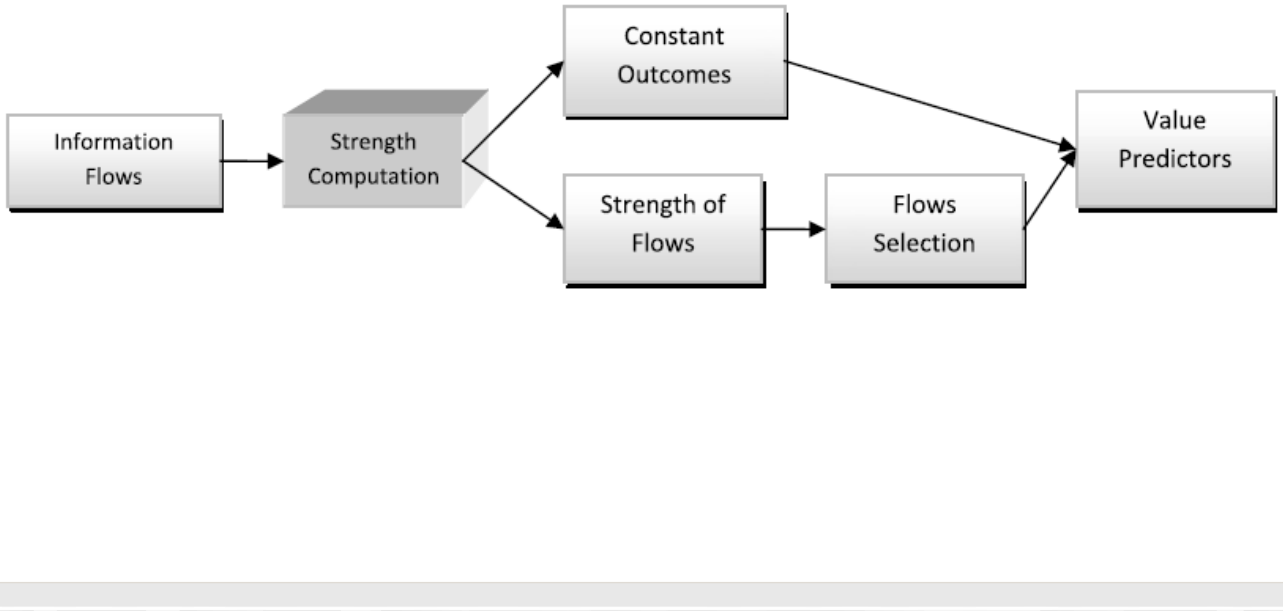


OVERVIEW OF STRENGTH OF FLOW COMPUTATION

- Computes the strengths of the identified dependences using information theoretic and statistical metrics applied on their associated values



DIFA DIRECTED DATA VALUE PREDICTION

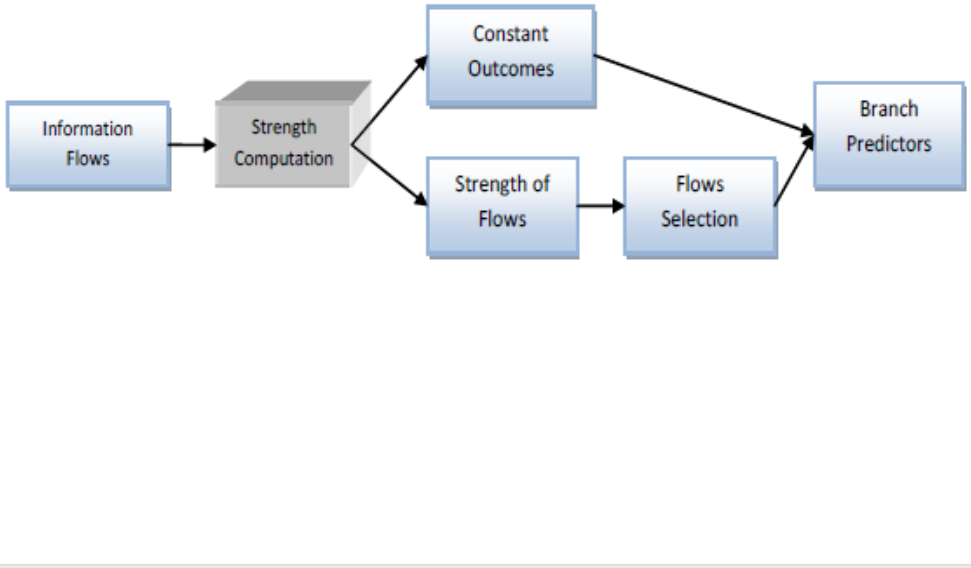


DIFA DIRECTED DATA VALUE PREDICTION

- Classify instructions
- Identify highly predictable instructions and selects the appropriate predictor to use
- Reduce the required hardware computations and achieves higher prediction accuracy
- Decrease the destructive impact of aliasing when it is present

DIFA DIRECTED INDIRECT BRANCH PREDICTION

- Identify the last instruction that updates the source operand of an indirect branch
- Indirect branch instructions not predictable using the regular BTB technique:
  - Access the prediction table by hashing the instruction pointer of the predicted instruction with the outcome of its source instruction.



DYNAMIC DEPENDENCE AND INFORMATION FLOW

Are dynamic dependences generally indicative of measurable information flow?

DYNAMIC DEPENDENCE AND INFORMATION FLOW

```
1   movl  $0, −12(%rbp)
2   jmp .L2
3   .L3:
4   movl  −12(%rbp), %ecx : (ins1), x value
5   movl  $1717986919, −36(%rbp)
6   movl  −36(%rbp), %eax
7   imull %ecx
8   sarl  %edx
9   movl  %ecx, %eax
10  sarl  $31, %eax
11  movl  %edx, %ebx
12  subl  %eax, %ebx
13  movl  %ebx, −28(%rbp)
14  movl  −28(%rbp), %eax
15  sall  $2, %eax
16  addl  −28(%rbp), %eax
17  movl  %ecx, %edx
18  subl  %eax, %edx
19  movl  %edx, −28(%rbp)
20  movl  −12(%rbp), %eax
21  subl  −28(%rbp), %eax
22  movl  %eax, −24(%rbp) : (ins2), a value
23  movl  −24(%rbp), %eax
24  addl  $1, %eax
25  movl  %eax, −16(%rbp) : (ins3), y value
26  addl  $1, −12(%rbp) : x++
27  .L2:
28  cmpl  $19, −12(%rbp)
29  jle .L3
30  movl  $0, %eax
31  popq  %rbx
32  leave
33  ret
```

```
for (int x=0; x<2
  a = x − x%5;
  y = a + 1;
}
```

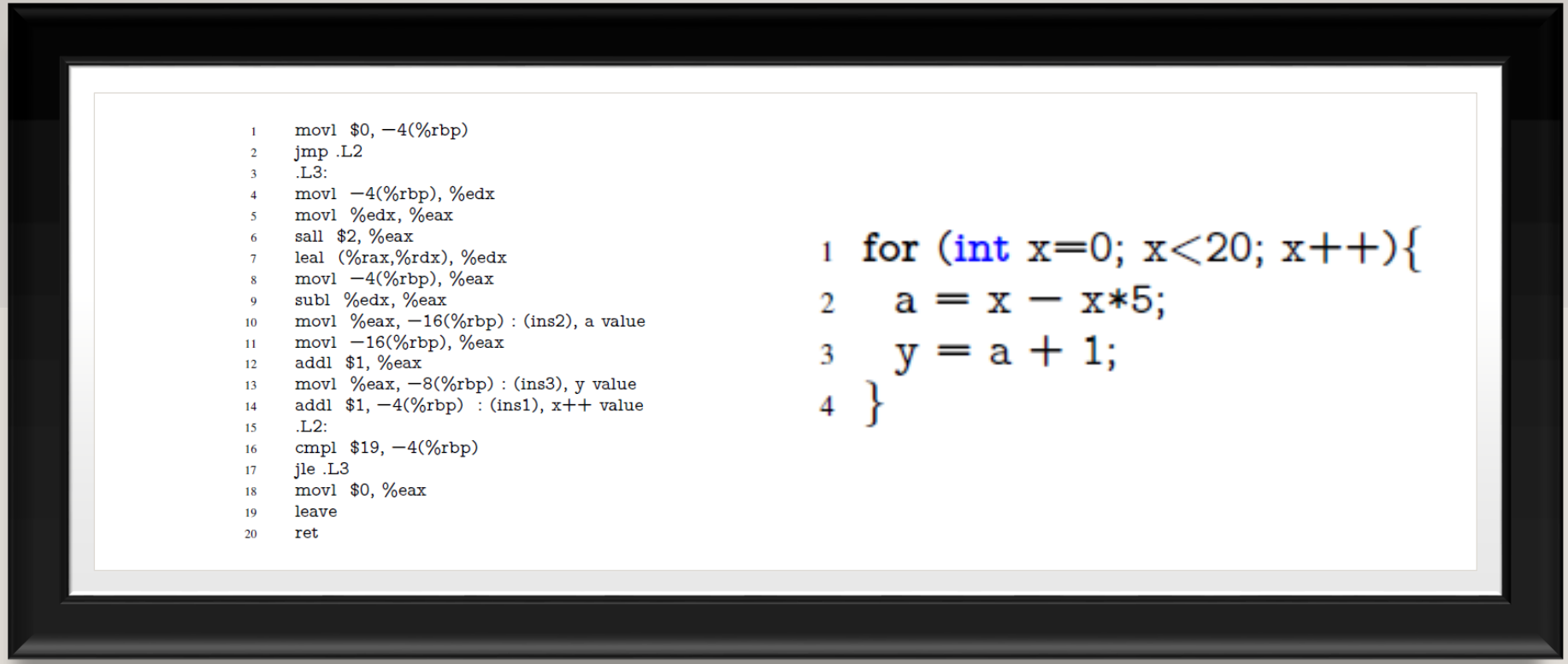
DYNAMIC DEPENDENCE AND INFORMATION FLOW

Strength of flows (ins1, ins2), (ins1, ins3) and (ins2, ins3) using normalized mutual information : 0.46, 0.46 and 1.0 respectively.

Learning that the outcome of ins2 is 0, an observer is not absolutely certain of the outcome of ins1, which could be 0; 1; 2; 3 or 4.

Similarly for the strength of flow (ins1, ins3).

Learning the outcome of ins3, we can infer the outcome of ins2 with 100% certainty.


1 movl $0, −4(%rbp)
2 jmp .L2
3 .L3:
4 movl −4(%rbp), %edx
5 movl %edx, %eax
6 sall $2, %eax
7 leal (%rax,%rdx), %edx
8 movl −4(%rbp), %eax
9 subl %edx, %eax
10 movl %eax, −16(%rbp) : (ins2), a value
11 movl −16(%rbp), %eax
12 addl $1, %eax
13 movl %eax, −8(%rbp) : (ins3), y value
14 addl $1, −4(%rbp) : (ins1), x++ value
15 .L2:
16 cmpl $19, −4(%rbp)
17 jle .L3
18 movl $0, %eax
19 leave
20 ret
1 for (int x=0; x<20; x++){
2 a = x − x*5;
3 y = a + 1;
4 }

DYNAMIC DEPENDENCE AND INFORMATION FLOW

- Strength of flows (ins1, ins2), (ins1, ins3) and (ins2, ins3) using normalized mutual information: 1.0 for all of them.
- Conclusion from the previous two examples: *the presence of dynamic dependence between two instructions is not a sufficient condition for the strong flow of information between them*

ZERO STRENGTH FLOWS

- Can we exploit paths that have zero strength flow to increase the effective instruction level parallelism (ILP)?

- Zero strength flows
  - Dependences where there are no correlation between the values of the source and those of the target.
  - Strength of a zero strength flow measured using eta coefficient, normalized mutual information and standard r is equal to 0.

ZERO STRENGTH FLOWS: CONCLUSIONS

- *The existence of dynamic dependences between two statements s1 and s2 is not a sufficient condition for the outcome produced by s1 to influence the outcome produced by s2.*

ZERO STRENGTH FLOWS AND ILP

- Select zero strength flows between distinct instructions where the source of flows are load instructions
- Analyze the predictability of the values of the source and the target of the selected flows

ZERO STRENGTH FLOWS AND ILP

- Selected source load instructions:
  - 40% selected for confident predictions
- Target of zero strength flows, with load instructions as the source of the flows
  - 86% are highly predictable
  - Account for around 3% of the totally dynamically executed instructions
- Conclusion: *Holding a highly predictable target instruction from execution since it is connected via zero strength flow with a non-predictable load instruction can negatively impact the execution performance*

Availability

https://github.com/wjghandour/dittanyTool/

QUESTIONS