Improving the performance of WCET analysis in the presence of variable latencies

XDD: a new structure to perform pipeline analysis

Zhenyu Bai, Hugues Cassé, Marianne De Michiel,Thomas Carle, Christine Rochange

CNRS - IRIT - University of Toulouse

COMPAS 2021

Introduction

- Critical real time system requires to ensure temporal properties.
- Worst Case Execution Time (WCET) on specific hardware
- Compute a safe upper bound of WCET by static analysis

WCET analysis: A typical structure

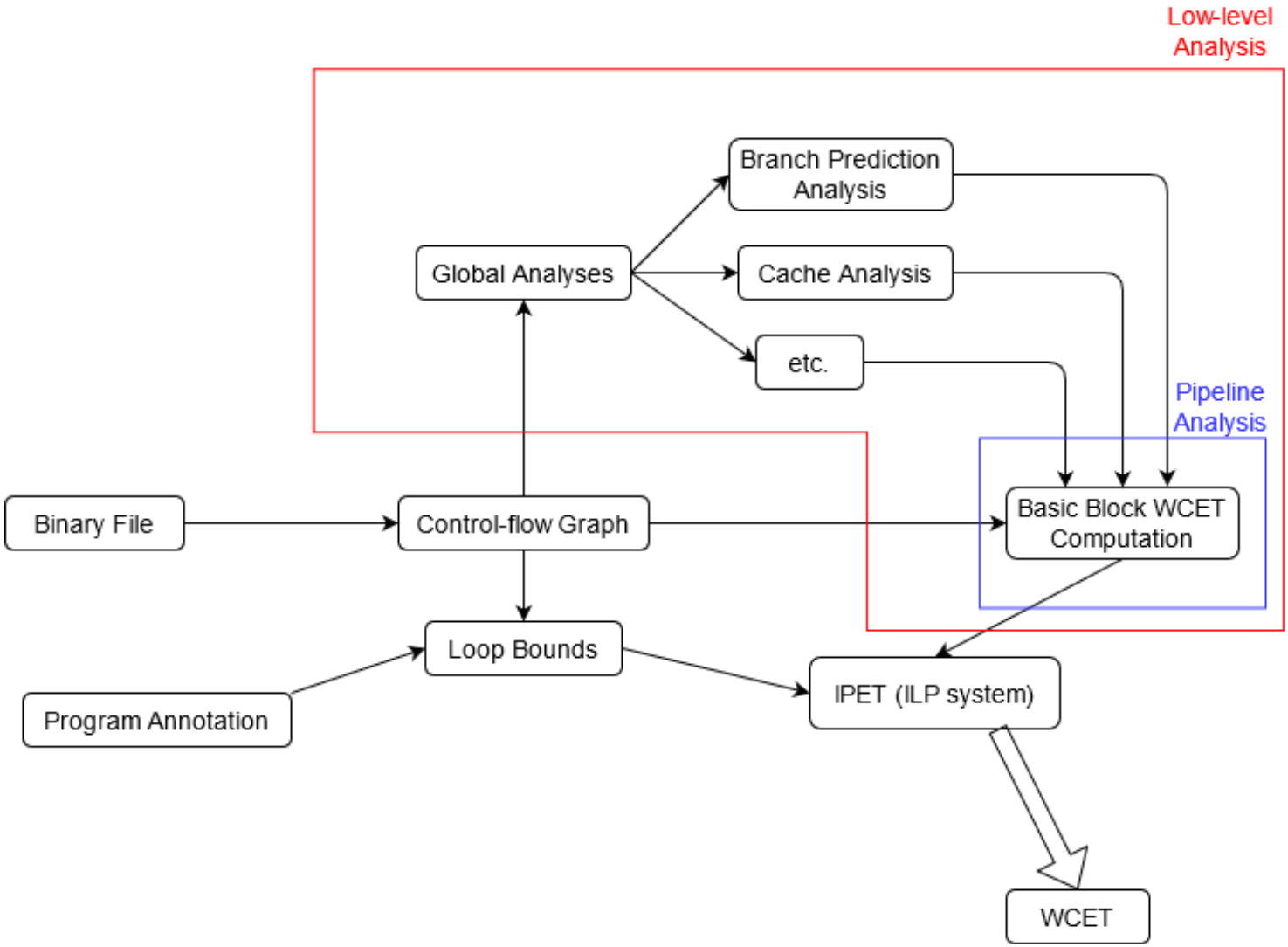
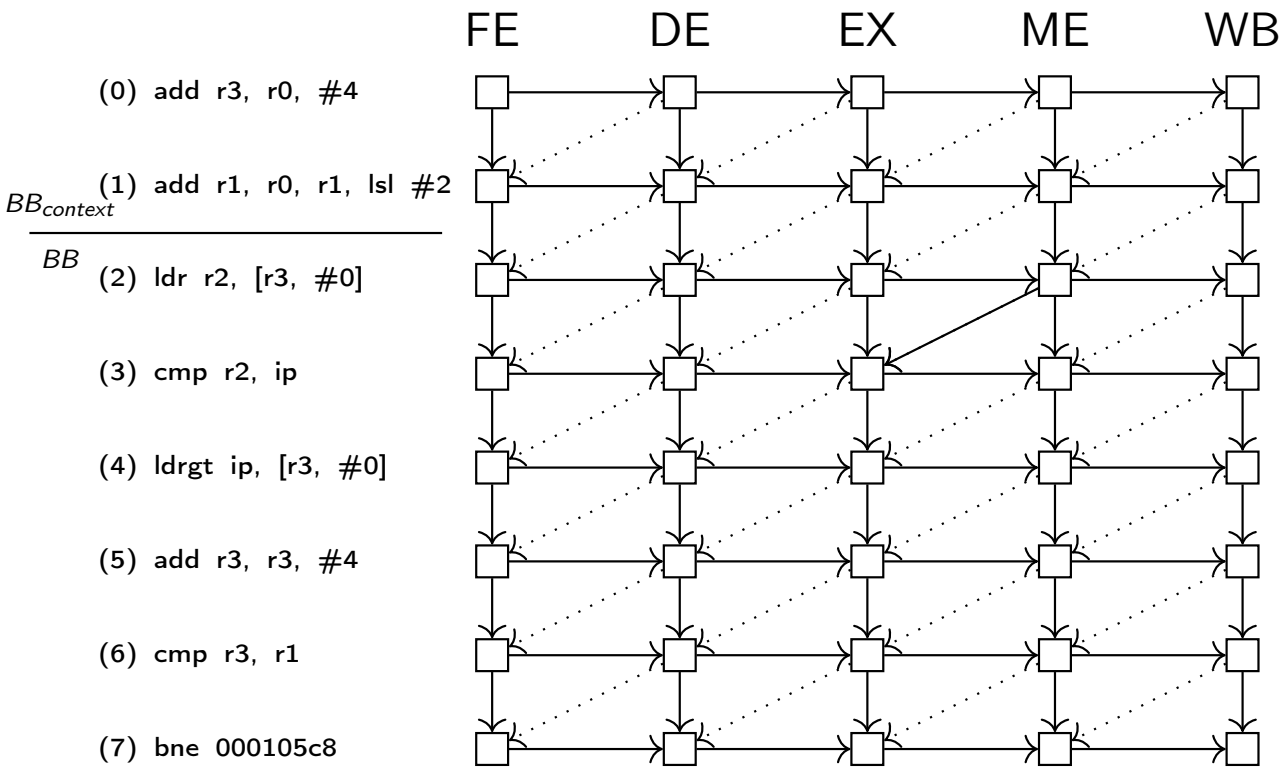


Figure: A typical WCET analysis structure

Execution Graph

- Execution Graph XG=(V_{XG}, E_{XG}) is a DAG representing the **dependencies** between pipeline stages;



Execution Graph

- $\delta = 1$: solid edge; $\delta = 0$: dotted edge
- λ : latency of node

Definition

The start time of node v, d_v^{start} can be computed by

$$d_v^{start} = \max_{w \to v \in E_{XG}} d_w^{start} + \delta(v \to w) \times \lambda(v)$$

Execution Graph

A basic exemple

$BB_{context}$

BB

(0) add r3, r0, #4

(1) add r1, r0, r1, lsl #2

(2) ldr r2, [r3, #0]

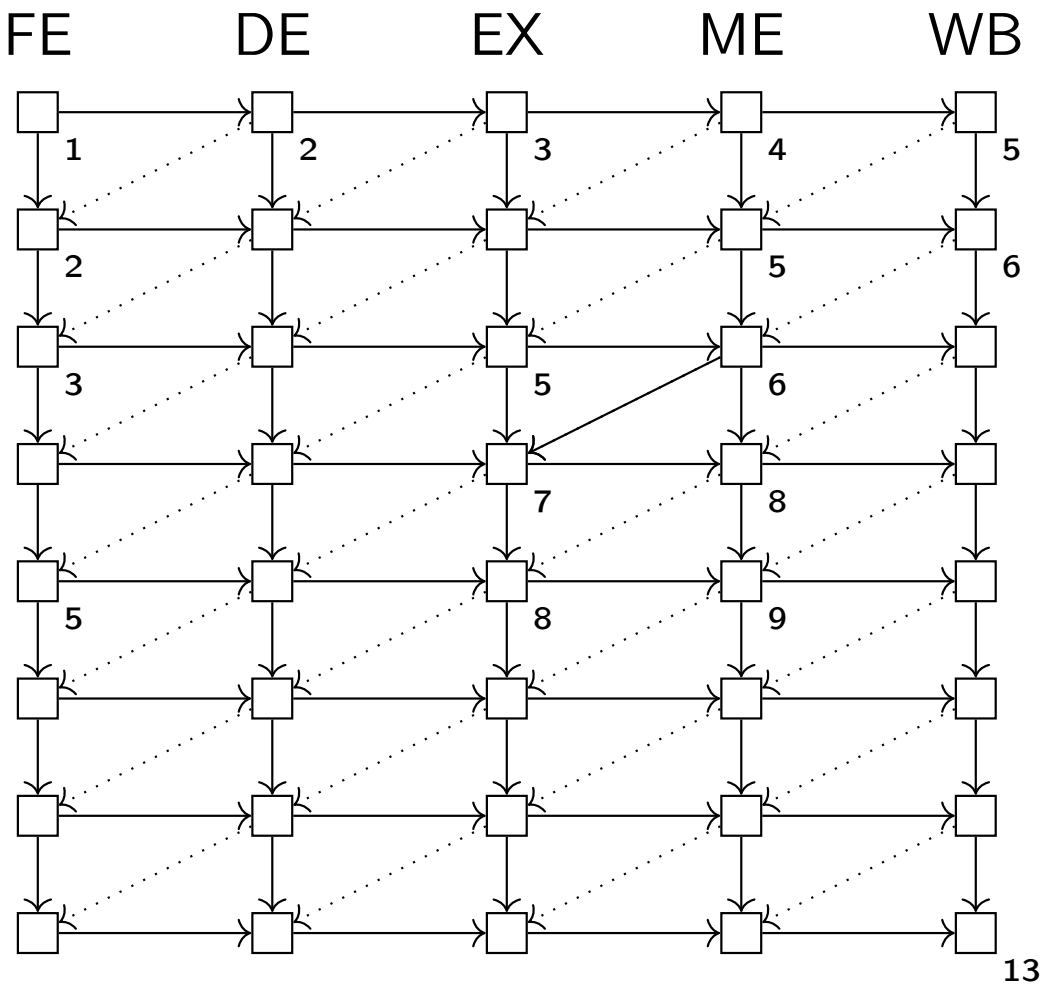
(3) cmp r2, ip

(4) ldrgt ip, [r3, #0]

(5) add r3, r3, #4

(6) cmp r3, r1

(7) bne 000105c8



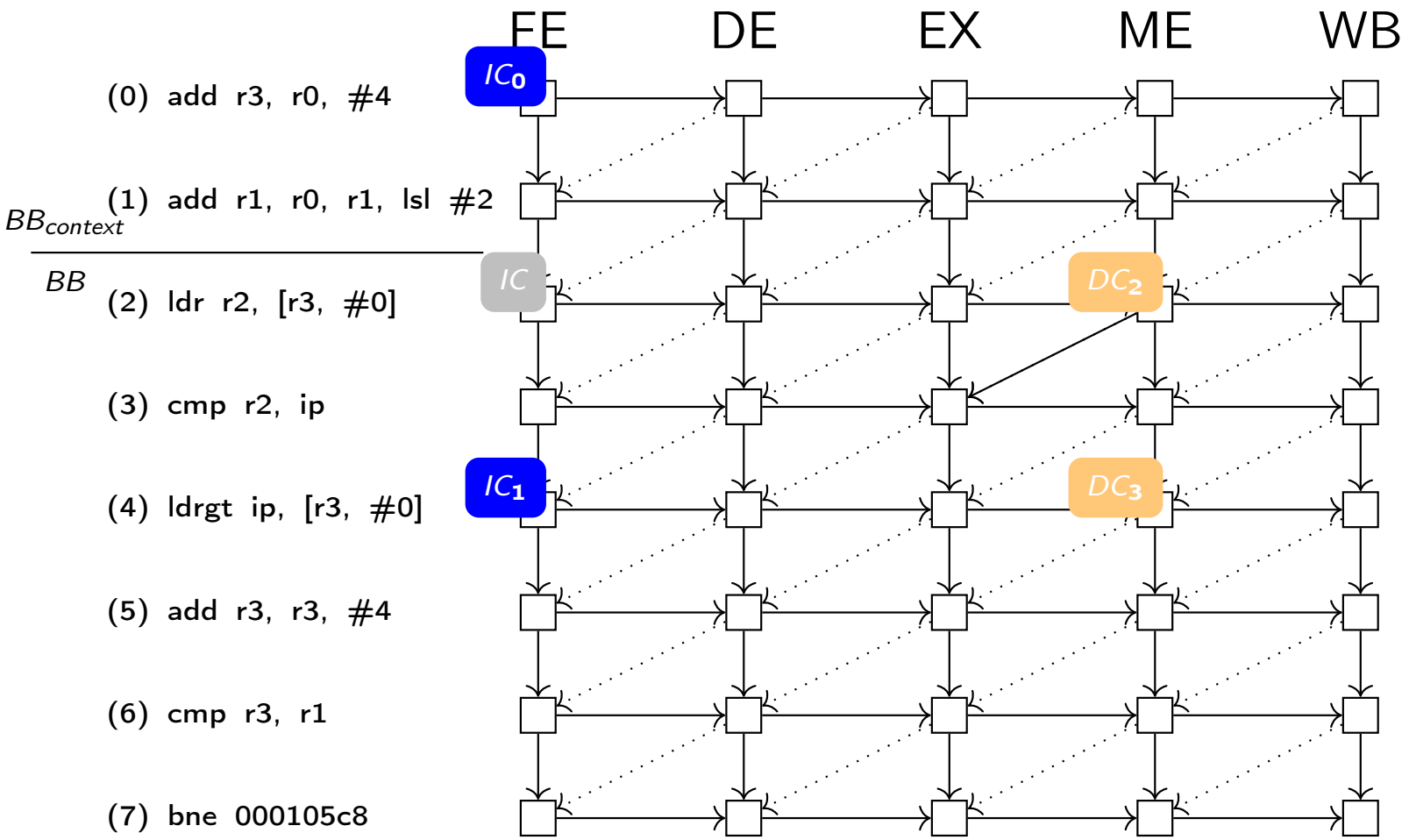
$$C_{BB-BB_{context}} = 13 - 6$$
$$= 7$$

Events

- Hardware effects involve timing variation
- Branch predictions, cache access, etc.
- Event $e \in \mathcal{E}$: binary variable representing the occurrence of a variable XG node processing time

Execution Graph

An exemple with events



BB execution time

Enumeration of BB execution time

$2^{|\mathcal{E}|} = 16$ configurations but only 5 times .

e_{IC_0}	e_{IC_1}	e_{IC_2}	e_{DC_3}	C_{BB}
*	0	0	0	6
*	0	0	1	16
*	0	1	0	16
*	0	1	1	25
*	1	0	0	15
*	1	0	1	24
*	1	1	0	16
*	1	1	1	25

BB execution time

BB execution time by decision tree

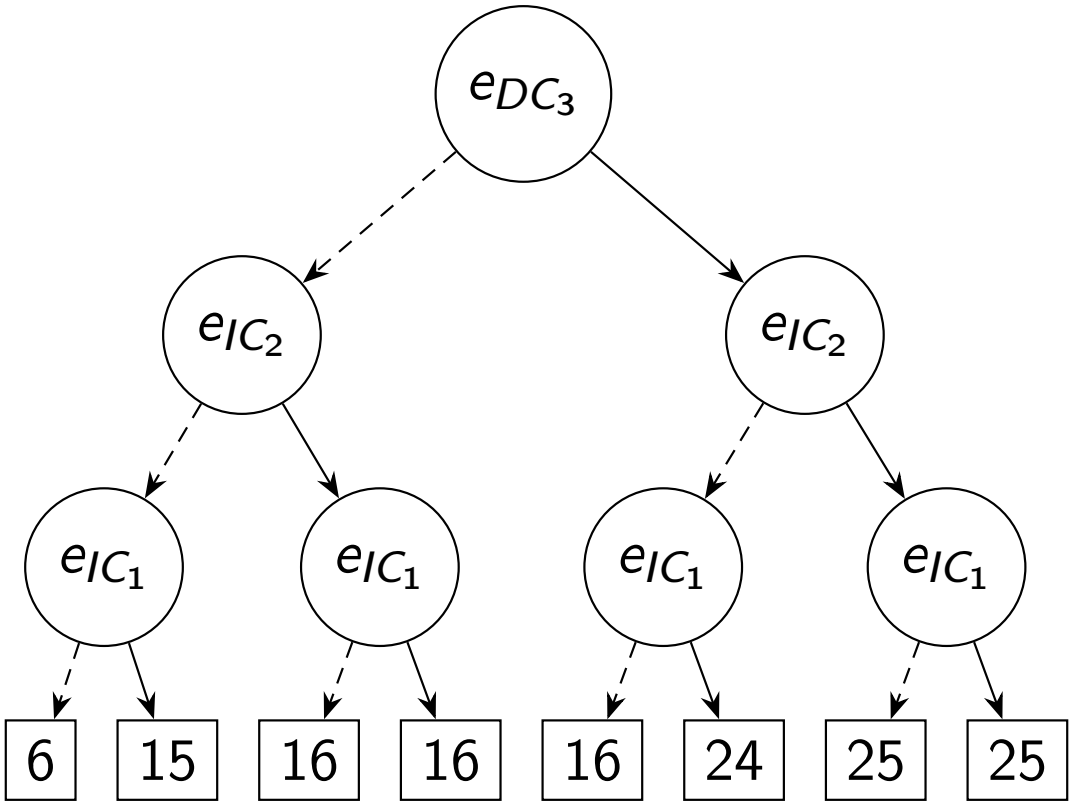


Figure: Full tree representation

BB execution time

XDD representation of BB execution time

Canonicity:

- The depth of events depends on a total order $(\mathcal{E}, \prec)$
- For a given total order $(\mathcal{E}, \prec)$ on events, there is a unique XDD tree representation.

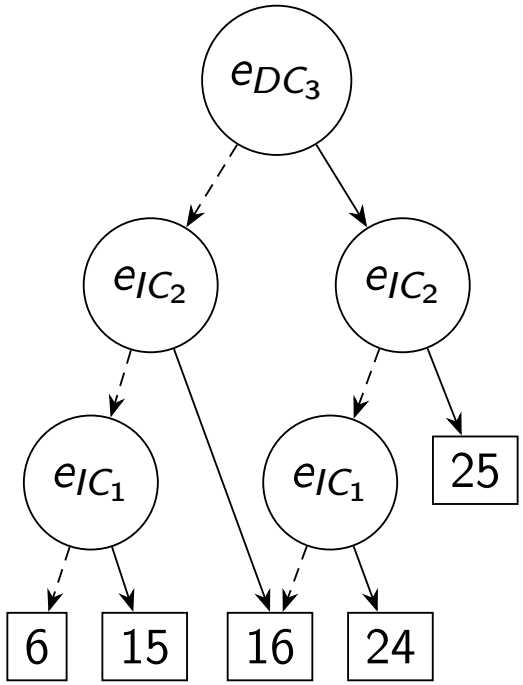


Figure: Execution Decision Diagram(XDD) representation

XDD in XG computation

- Compatible with XG:

$$d_v^{start} = \max_{w \to v \in E_{XG}} d_w^{start} + \delta(v \to w) \times (\lambda(v))$$

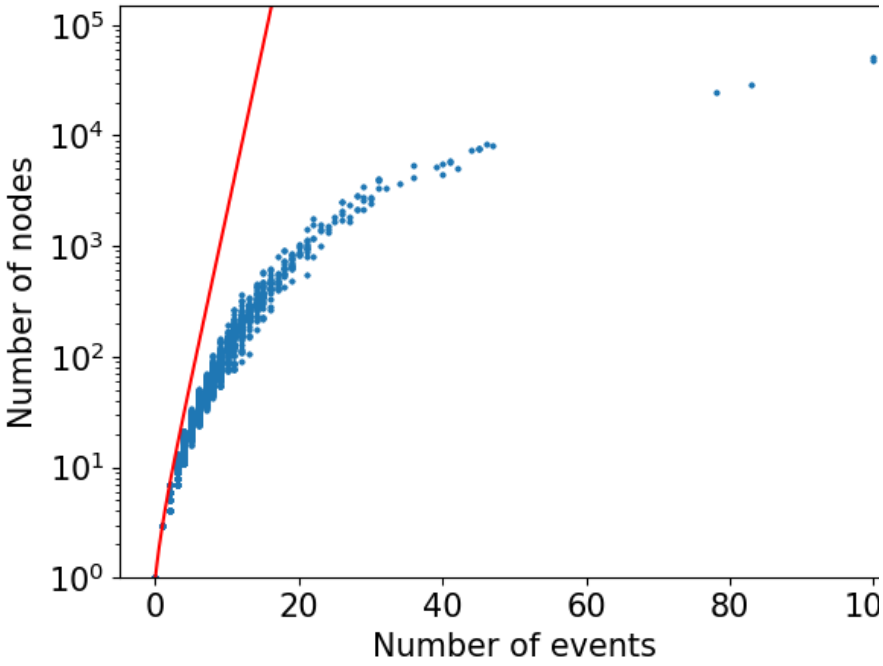
becomes :

$$d_v^{\#} = \bigoplus_{w \to v \in E_{XG}} d_w^{\#} \otimes (\lambda^{\#}(w) \times \delta(w \to v))$$

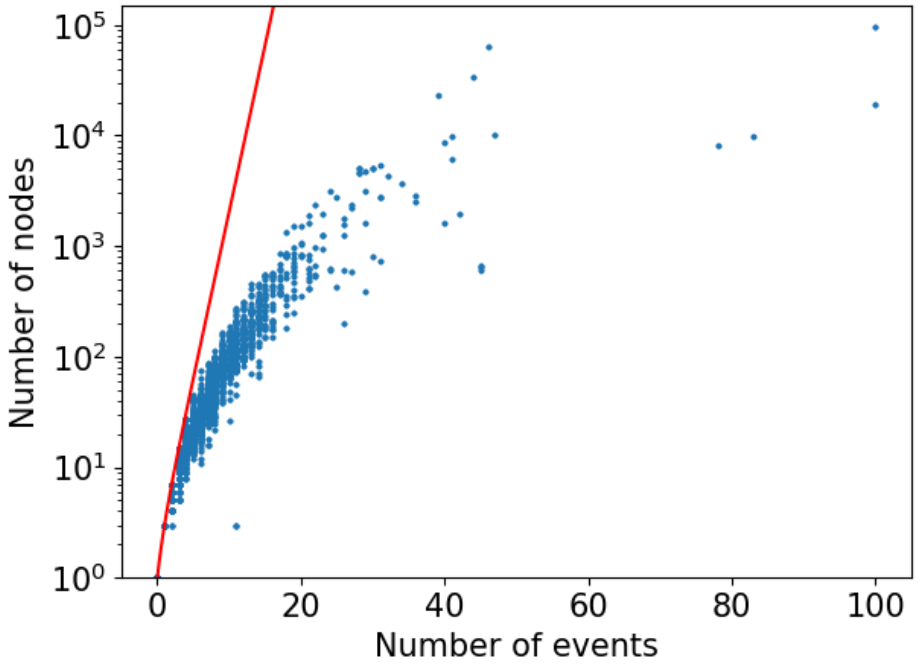
$(\oplus \simeq \max; \otimes \simeq +)$

- Represent all possible times at each XG node
- Analysis in one pass instead of $2^{|\mathcal{E}|}$ passes

Experiments



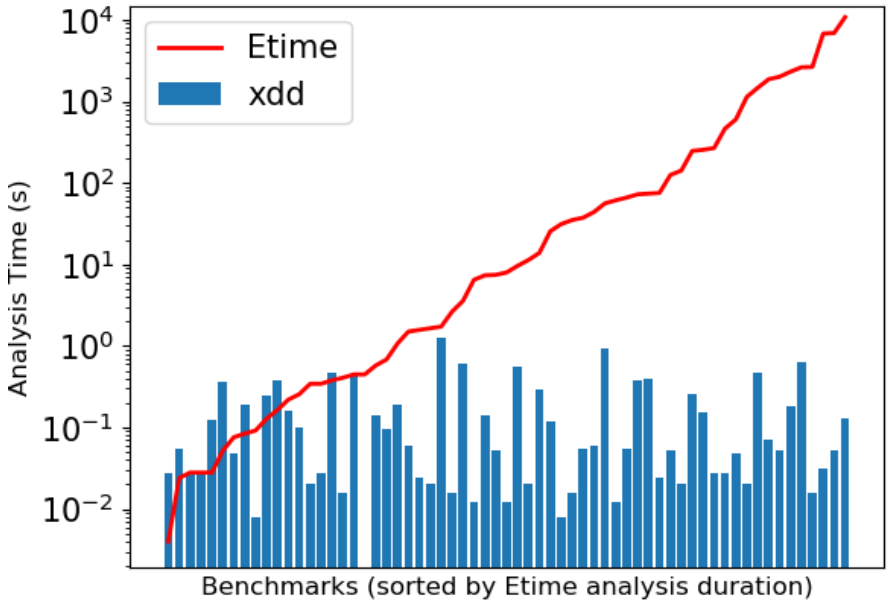
(a) Simple Architecture



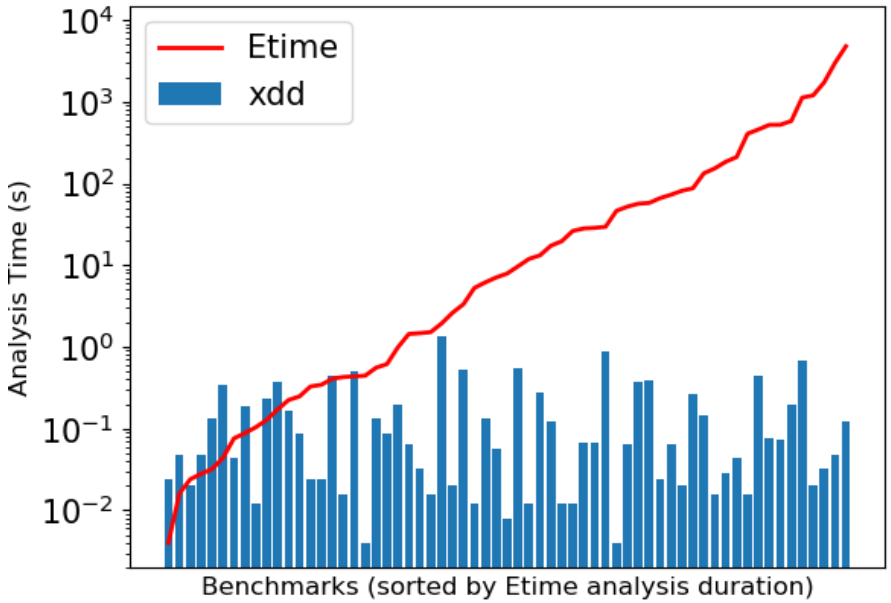
(b) Complex Architecture

Figure: Number of nodes with respect to the number of events

Experiments



(a) Simple architecture



(b) Complex architecture

Figure: Analysis time

Conclusion

XDD

- compact structure to represent timing variation;
- offers an efficient way to perform pipeline analysis with XG;
- Performance of XDD relies on the pipeline absorption effect;
- Experiments show the impressive compress ratio;
- Up to 1000 times faster than current exhaustive approach.

Future Works

- Using XDD to model out-of-order pipeline;
- Experiment on different hardware architecture;