Instruction Duration Estimation by Partial Trace Evaluation

Matteo Corti, Thomas Gross

ETH Zürich

Goal

- Tight, conservative **WCET** approximation.
- **Large non–safety-critical applications**.
- **Modern CPUs:**
  - complex pipelines, caches and branch prediction
  - preemptive system

- ERCO:
  - ahead-of-time Java to native compiler
  - semantic analyzer
  - **instruction duration estimator**

Input

- Assembler file with **semantic information**
- Control flow graph
- **Maximum number of iterations for each block**
- Dynamic and static call targets (call graph)
- False paths

Instruction Duration

- Goal: compute the duration of the single instructions.
- The duration depends on the context.
- Limited computational context:

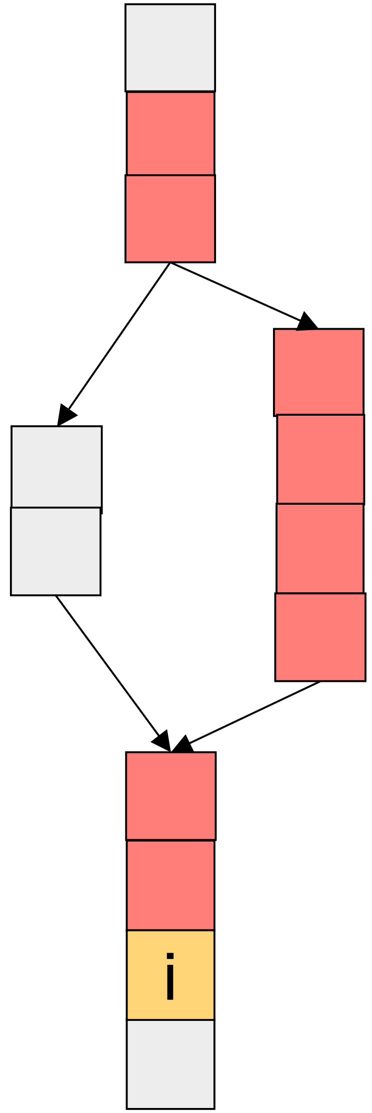
We assume that the effects on the pipeline and caches of an instruction fade over time.

Locality: Limited Context

- Partial trace / context:

  the last *n* instructions before the instruction *i* on a given trace

- n is determined experimentally (50-100 instructions)



Estimated Instruction Duration

- For every partial trace:
  - CPU behavior simulation
  - duration according to the context
- We account for all the incoming paths (contexts) according to their iteration counts.
- Block duration = ∑ instruction durations
- WCET = longest path (CFG)

Results: Small Kernels

Benchmark	Estimated	Measured	
Division	$1.545 \cdot 10^{9}$	$1.400 \cdot 10^{9}$	10.351%
Jacobi	$1.075 \cdot 10^{10}$	$8.788 \cdot 10^{9}$	22.351%
MatrixInversion	$1.553 \cdot 10^{9}$	$1.419 \cdot 10^{9}$	9.402%
MatrixMult	$2.732 \cdot 10^{9}$	$2.667 \cdot 10^{9}$	2.448%

- Increasing the partial trace length does not change the results

Results: Applications

Benchmark	Measured	Observed	
_201_compress	$9.45 \cdot 10^{9}$	$1.11 \cdot 10^{10}$	117%
javalayer	$2.67 \cdot 10^{9}$	$1.30 \cdot 10^{10}$	487%
scimark2	$2.47 \cdot 10^{10}$	$1.42 \cdot 10^{11}$	579%

Conclusions

- Asymptotically **linear time estimation** of the instruction duration.
- Can be adapted to different CPU/architectures and semantic analyzers.
- **Working prototype** delivering fair estimations.
- Room for improvement in the CPU simulation.