

Dec 04, 2017 – IBM PL DAY

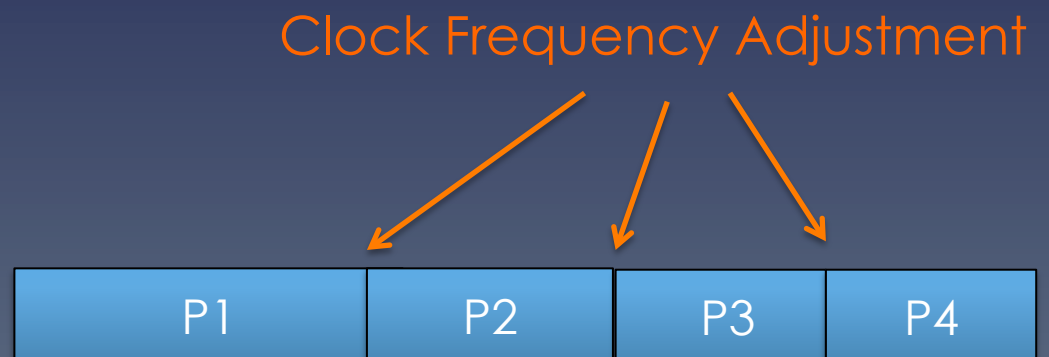
Applying Decision Trees to Reduce Energy-Delay Product

Meng-Chieh (Joe) Chiu
and Eliot Moss

{joechiu,moss}@cs.umass.edu
University of Massachusetts, Amherst

Introduction

- Goal: real-time energy-delay product saving
 - For statically compiled executables
 - Low run-time overhead
 - High ED saving



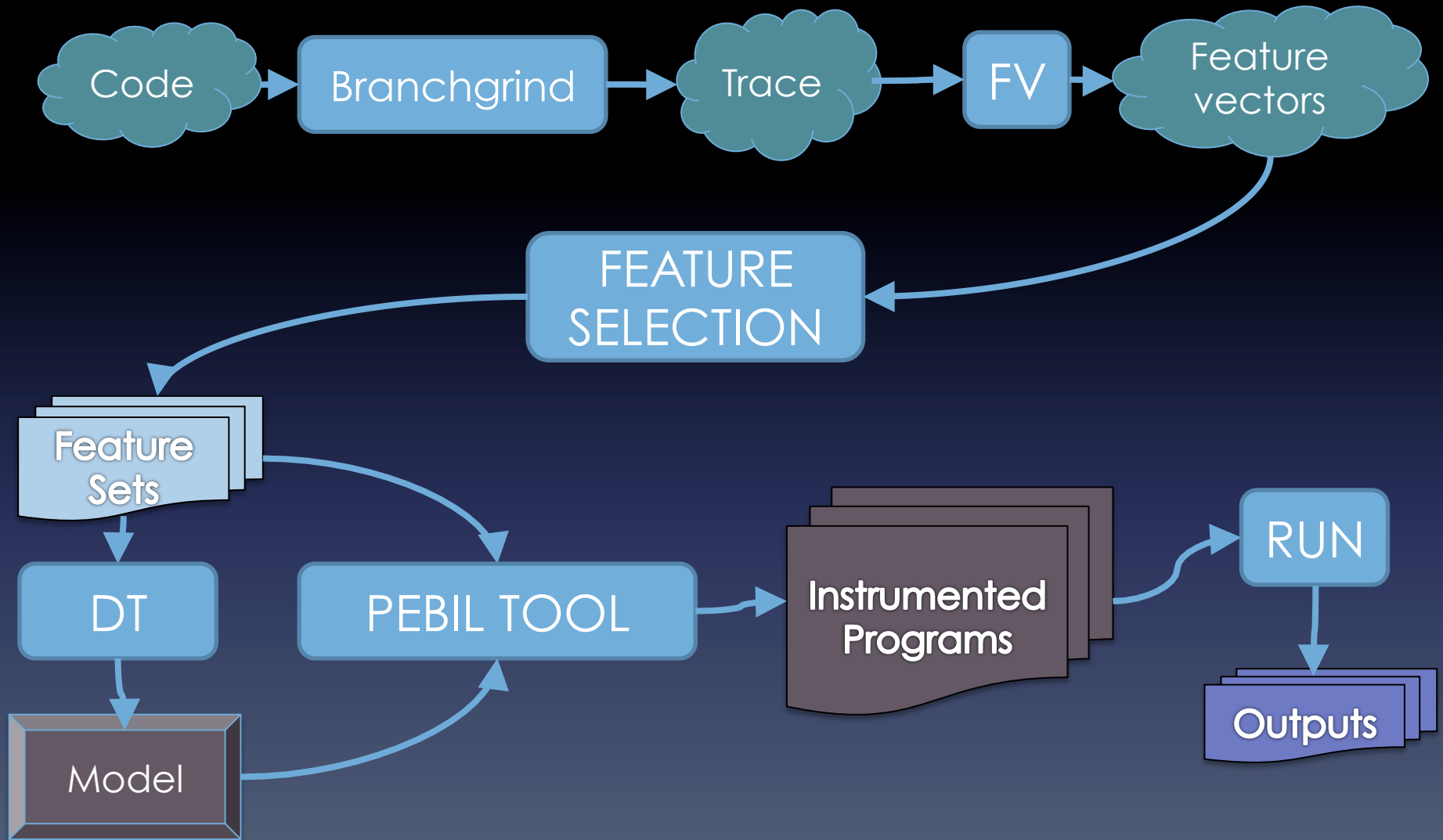
Why

- Why not?

The Challenge

- Adjustment needs information
- Too much info: high overhead
- Too little info: low quality

Workflow



Energy-delay product approximation

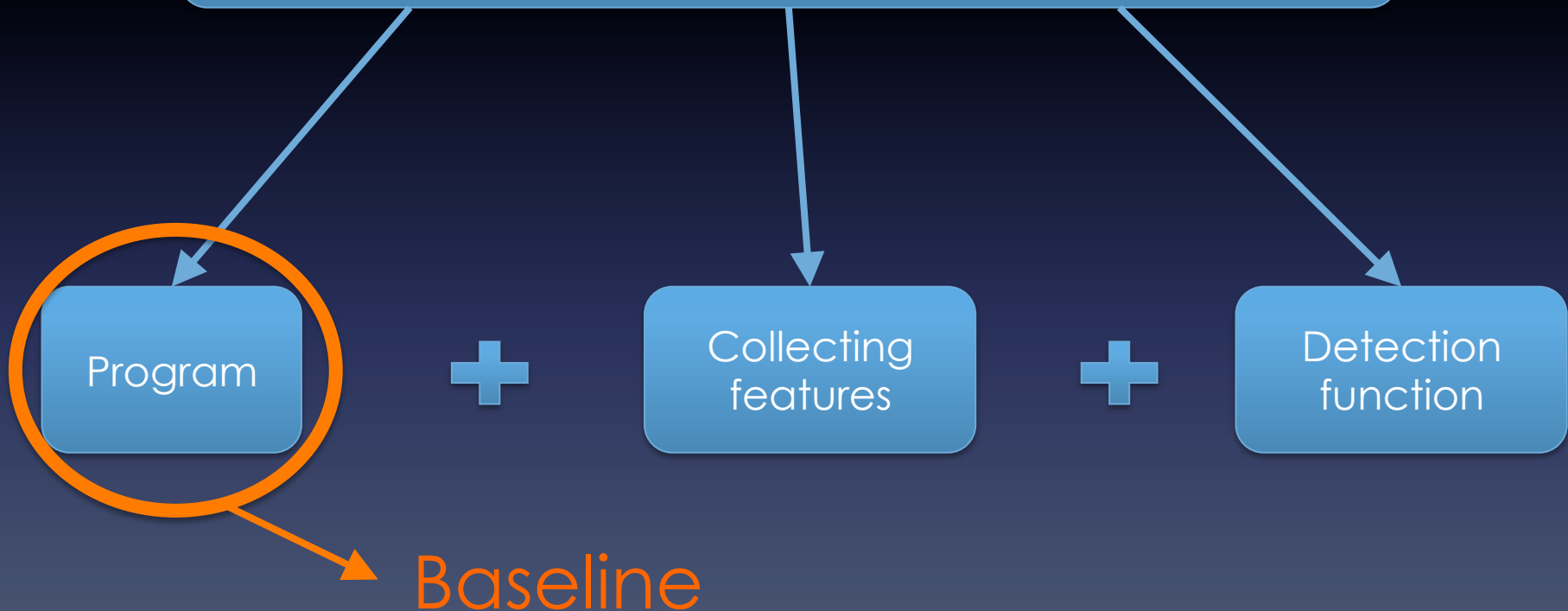
- $F \propto (V - V_{th})^\alpha / V$
 - V_{th} is about 0.4v and α is about 1.4
- $V = f / k + V_{th}$ where $k = 0.8$
- $ED \propto V^2 \times t_f^2$
 $\propto (1.25f + 0.4)^2 f \times t_f^2$
 $= (1.25f + 0.4)^2 (f + M/C)^2 / f$

Runtime Overhead

- SPEC CPU2006 benchmarks
- **5 runs** of the instrumented program, for each input
- Baseline: **5 runs** (no instrumentation)

Run-time Overhead

Running time of the instrumented program



Run-time Overhead

Benchmark	Inst 6	Inst 8
bzip2-ref	1.9%	3.1%
bzip2-train	0.3%	0.5%
gcc-ref-1	1.0%	1.8%
gcc-ref-2	1.2%	0.8%
gcc-ref-3	1.3%	0.8%
gcc-train	2.1%	1.4%
h264ref-ref	0.7%	0.1%
Average	1.2%	1.2%

The Quality (real time)

- 9 different parameter sets ED approximation, varying k and V_{th}
 - $V = f / k + V_{th}$
- Score matrix: (ED - optimal) / optimal
 - Optimal result from all possible available clock frequency settings

The Quality (real time)

Settings		Inst 6	Inst 8
k	v _{th}		
0.6	0.2	30.5%	30.0%
0.6	0.4	27.1%	25.3%
0.6	0.6	23.2%	20.0%
0.8	0.2	29.4%	28.5%
0.8	0.4	24.5%	21.9%
0.8	0.6	18.4%	13.4%
1.0	0.2	28.3%	27.0%
1.0	0.4	21.7%	18.0%
1.0	0.6	10.4%	2.6%
Average		23.7%	20.7%

Our model is better than Fixed(232.8%) and LR(182.9%)

Conclusion

- A method for real-time clock frequency adjustment that results in near-optimal energy-delay product
 - Good quality
 - Minimal performance impact

Acknowledgments

- NSF Grant CCF-1320498
- Eliot Moss

Thanks!

joechiu@cs.umass.edu