Simple Profile Rectifications Go A Long Way

—Statistically Exploring and Alleviating

the Effects of Sampling Errors for Program Optimizations

Bo Wu   Mingzhou Zhou   Xipeng Shen

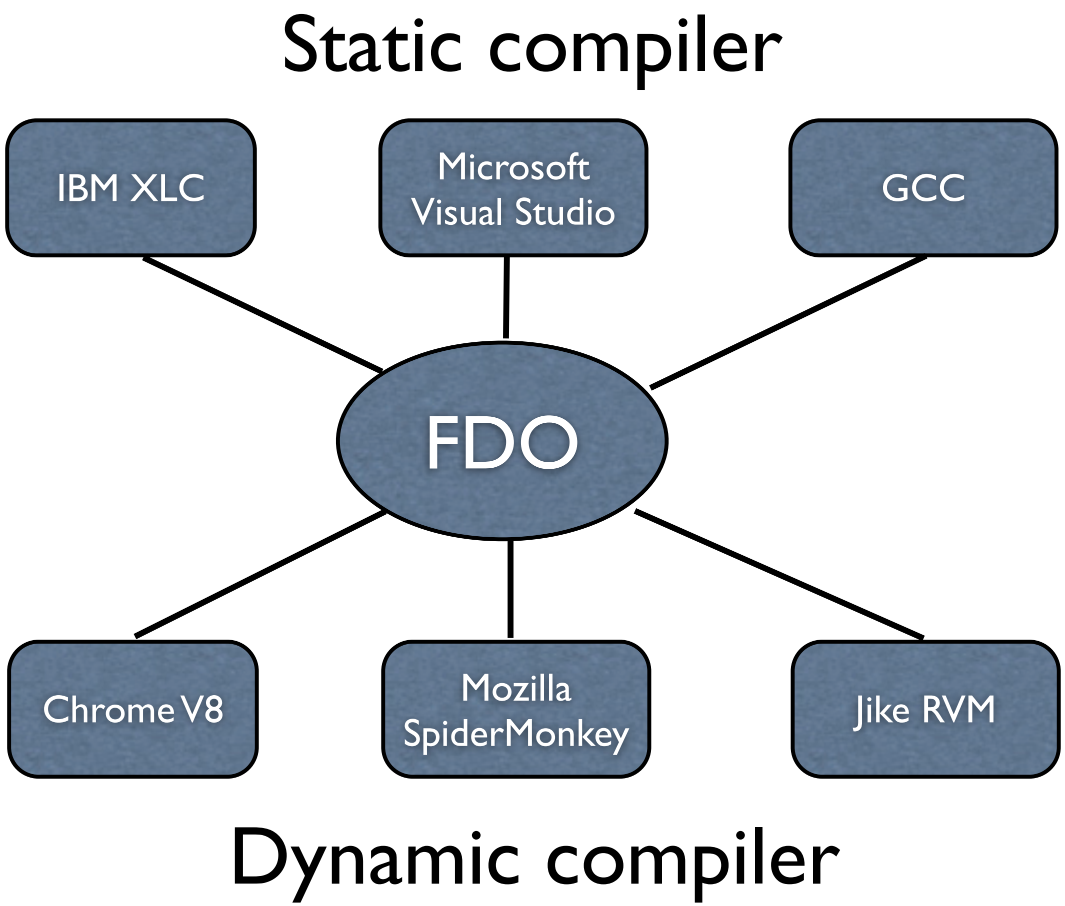
The College of William and Mary

Yaoqing Gao   Raul Silvera   Graham Yiu

IBM Toronto Lab

WILLIAM & MARY

FDO (feedback-driven optimizations)



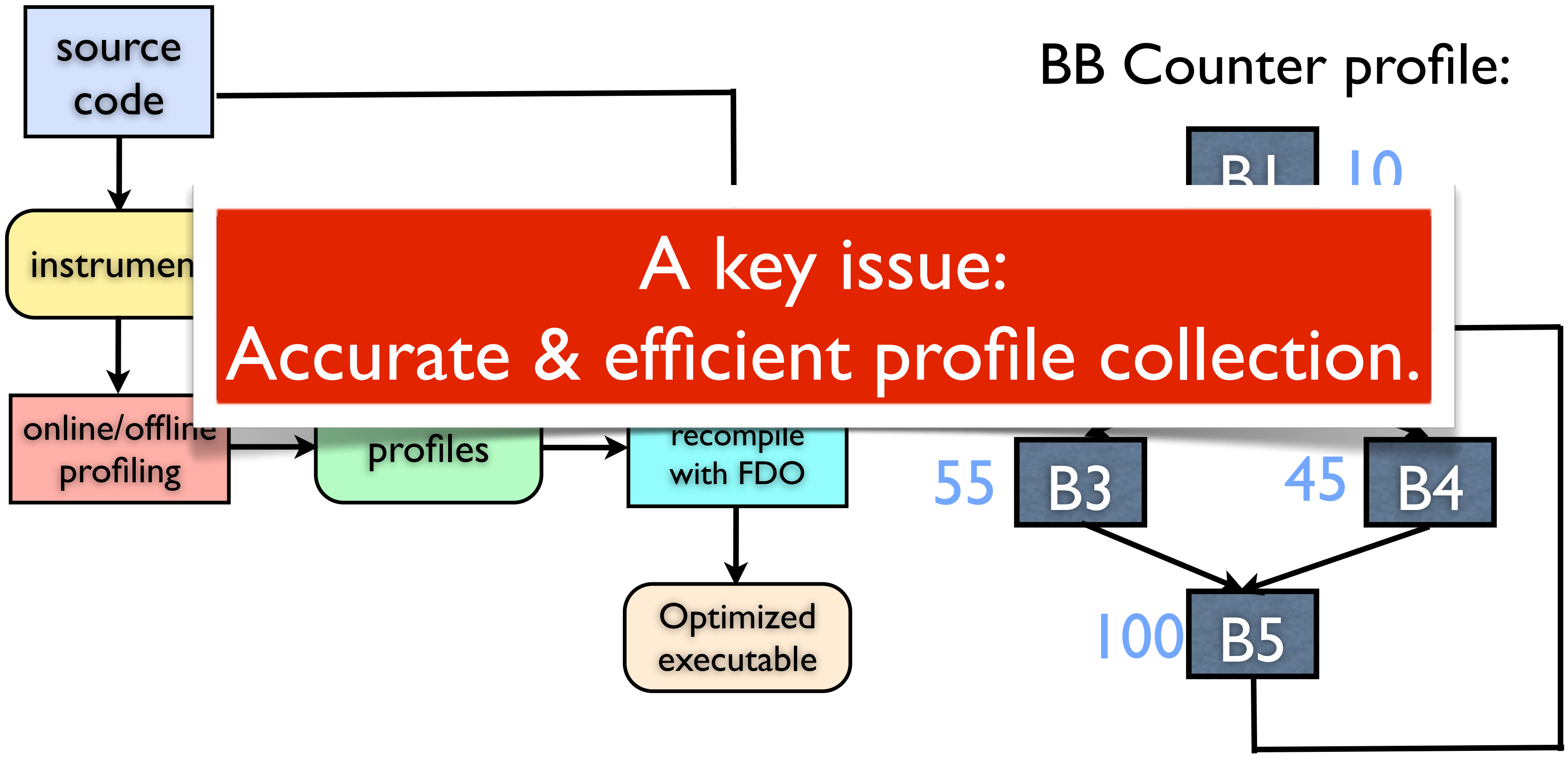
Guiding Opt:

- Code layout enhancement
- Function inlining
- Loop unrolling
- Type specialization

… …

WILLIAM & MARY

Workflow of FDO



A key issue:
Accurate & efficient profile collection.

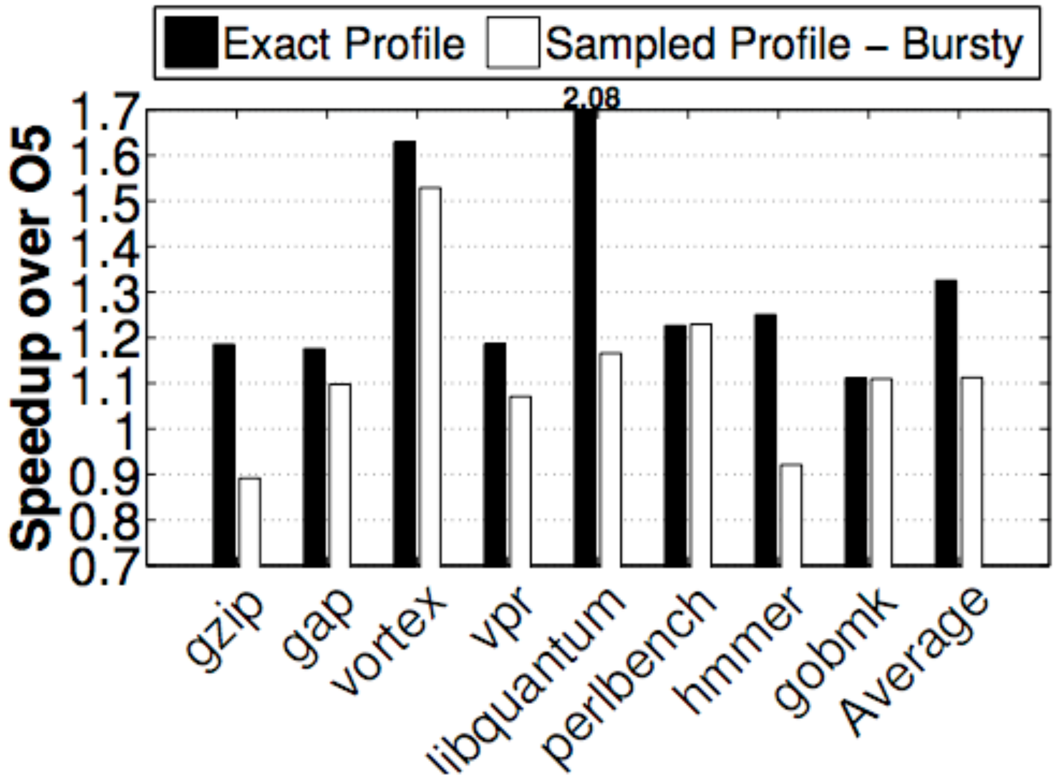
Sampling for FDO

- Take samples rather than profile the whole execution
  - A common approach to efficiency
    - Jikes RVM, J9, V8, Google cluster app. sampling, etc.

- Many efforts for efficient profiling with good accuracy
    - [Arnold and Ryder, PLDI'01], [Bond and McKinley, MICRO'05], [Arnold and Grove, CGO'05], ...

WILLIAM & MARY

FDO speedups



5% sampling rate

1/3 benefits

Substantial gap from what exact profiles provide

Goal of This Work

A new angle to attack the problem:
Rectify sampling errors for FDO.

- Questions to answer
  - ❖ Do higher sampling rates really give more useful profiles?
  - ❖ What types of errors in a sampled profile are essential?
  - ❖ Can they be easily fixed?

WILLIAM & MARY

Outline

- Measuring the relations between sampling and FDO
- Analyzing critical sampling errors
- Statistical profile rectification

Impact of sampling on FDO

Common Perception

Sampling rate ↑ ⇨ Profile accuracy ↑ ⇨ Usefulness for FDO ↑

WILLIAM & MARY

A systematic measurement

(4 levels, 8 dimensions, 7680 runs)

Levels	Dimensions	Variations	Description
sampling	methods	2	Bursty, Uniform
	frequencies	12	6 for Bursty, 6 for Uniform
system	compilers	2	XLC, GCC
	platforms	2	Intel Xeon & IBM POWER7
workload	benchmarks	8	SPEC CPU2000 & CPU2006
	inputs	4	1 train input, 3 ref inputs
noise avoidance	repetitions	10	number of repetitive runs per setting

- BB counter profile for XLC, edge profile for GCC
  - **Bursty sampling**: Sample *m* BBs for every *n* BB executions
  - **Uniform sampling**: Sample 1 instruction every *n* dynamic instructions
- Using C/C++ compilers
  - Sophisticated FDO optimizers
  - Better noise control that JIT
- 8 SPEC benchmarks most sensitive to FDO

WILLIAM & MARY

A systematic measurement

(4 levels, 8 dimensions, 7680 runs)

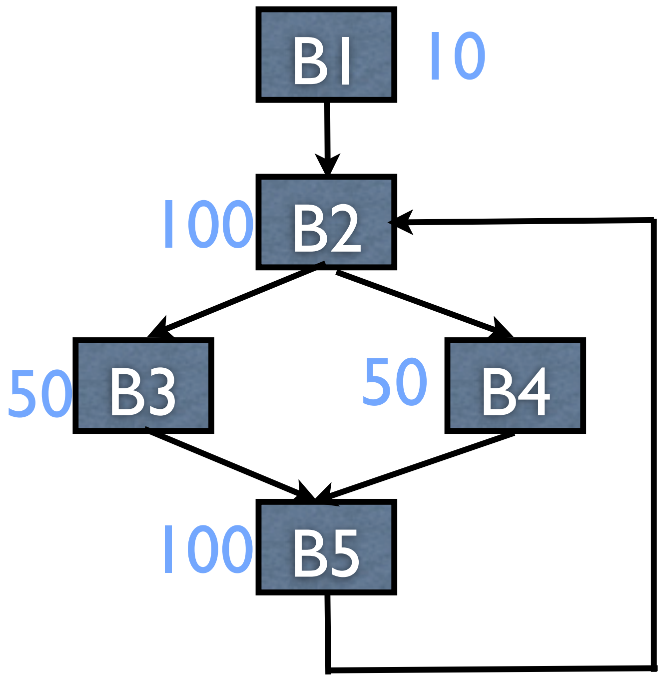
Levels	Dimensions	Variations	Description
sampling	methods	2	Bursty, Uniform
	frequencies	12	6 for Bursty, 6 for Uniform
system	compilers	2	XLC, GCC
	platforms	2	Intel Xeon & IBM POWER7
workload	benchmarks	8	SPEC CPU2000 & CPU2006
	inputs	4	1 train input, 3 ref inputs
noise avoidance	repetitions	10	number of repetitive runs per setting

- Sampling frequencies
  - 1/1000, 10/1000, 50/1000, 100/1000, 200/1000, 400/1000

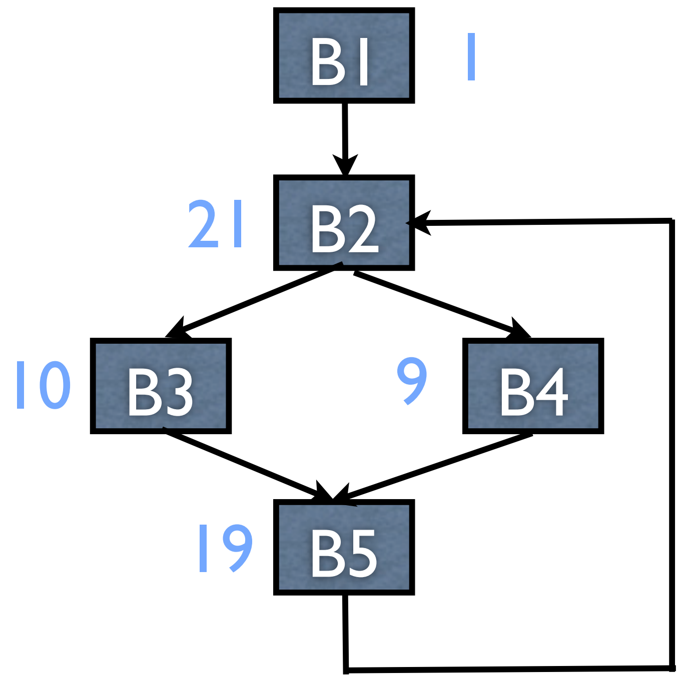
WILLIAM & MARY

Profile accuracy definition

Exact BB counter profile:



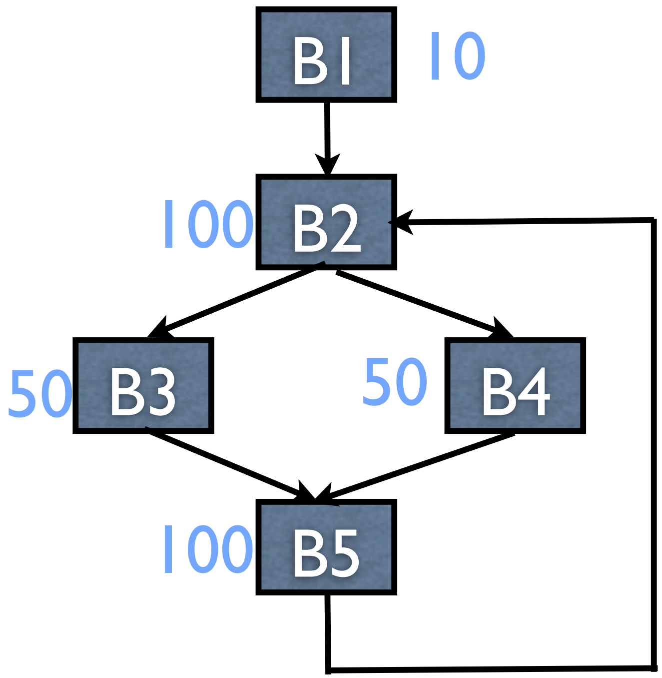
Sample BB counter profile:



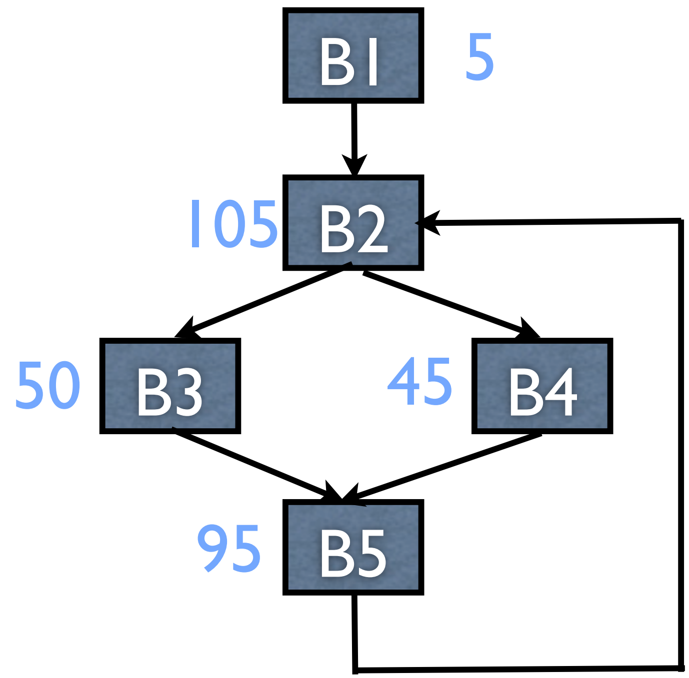
(e.g., sampling rate: 1/5)

Profile accuracy definition

Exact BB counter profile:

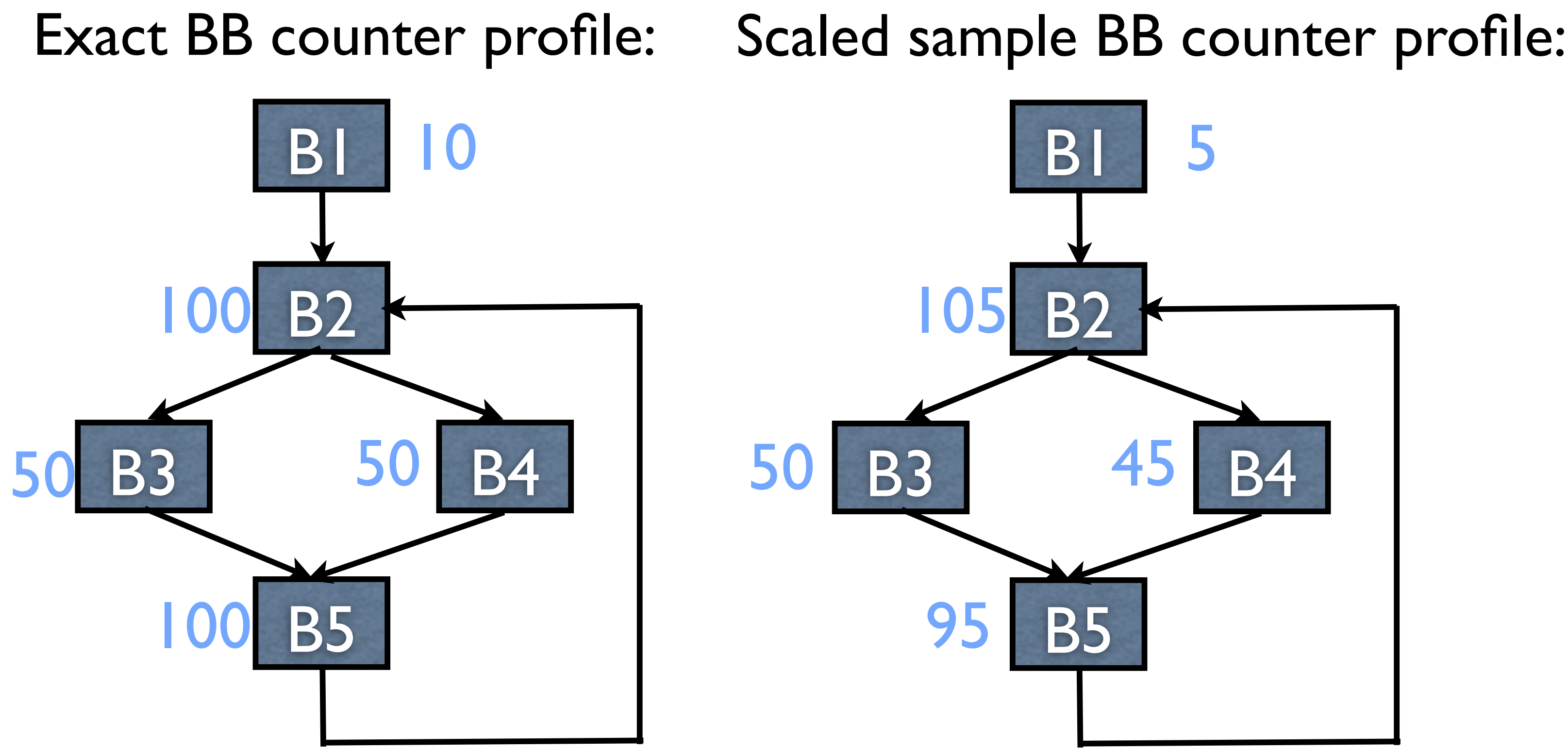


Scaled sample BB counter profile:



(e.g., sampling rate: 1/5)

Profile accuracy definition



Accuracy of B1: 1 - |10-5| /10 = 0.5

Unweighted accuracy: avg of all BB's accuracy

Weighted accuracy: weighted with the exact counter values

Metric for correlations

- Spearman's rank correlation coefficient
  - ❖ Quantifying the monotonic relationship

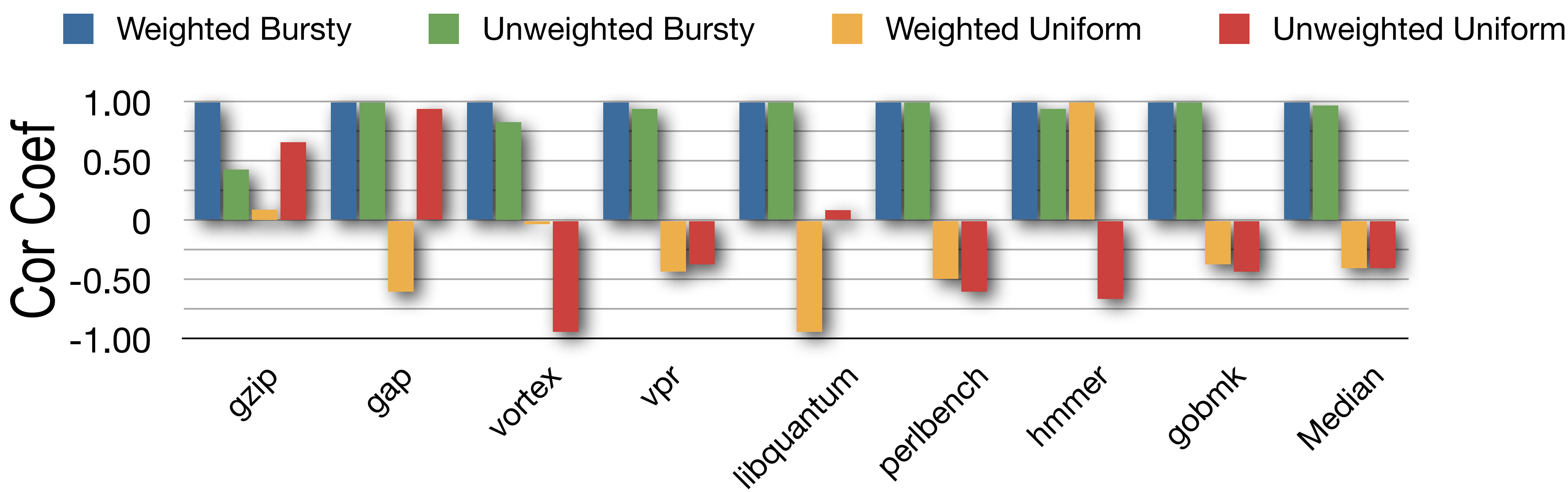
(sample rate 1, accuracy 1)
(sample rate 2, accuracy 2)
(sample rate 3, accuracy 3)
... ...

x_i: rank of sample rate i
y_i: rank of accuracy i

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

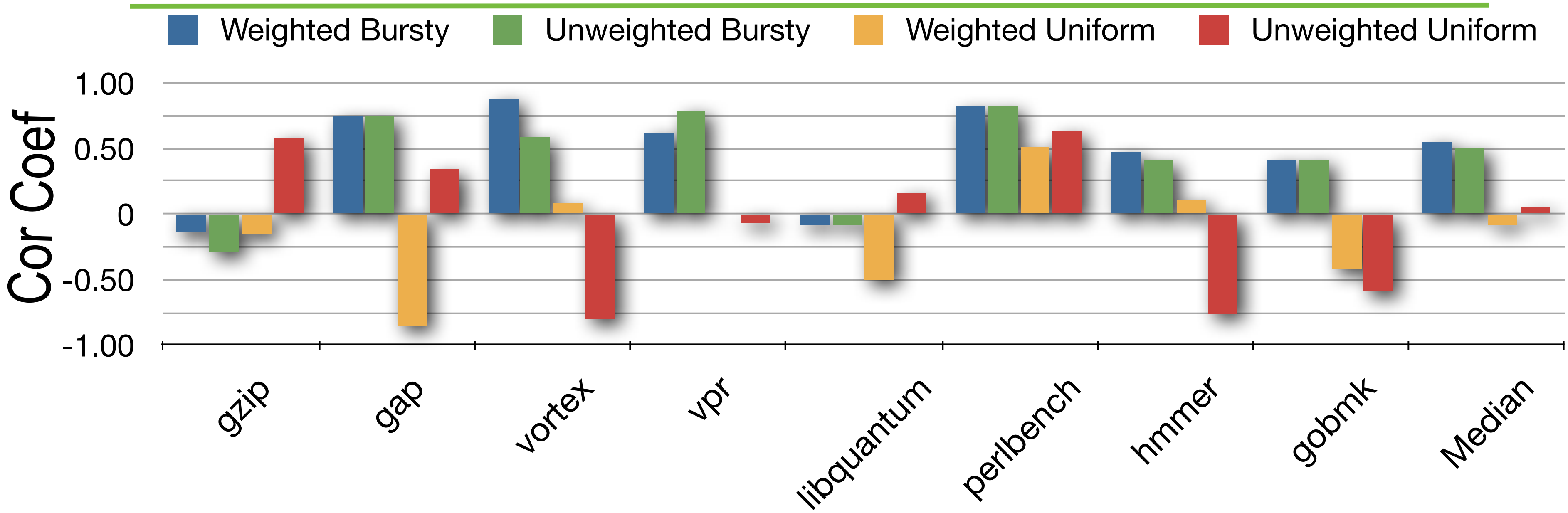
WILLIAM & MARY

Sampling rates vs. Profile accuracy



- Bursty sampling: Higher rates => Better accuracy
- Uniform sampling: not the case.

Profile accuracy v.s. FDO speedup



- Profile accuracy weakly correlates with speedup.
  - ❖ Hypothesis 1: the FDO fails to exploit profiles effectively
    - ✤ Unlikely as it gives best performance on exact profiles.
  - ❖ Hypothesis 2: Some errors hurt speedups but not accuracies much.

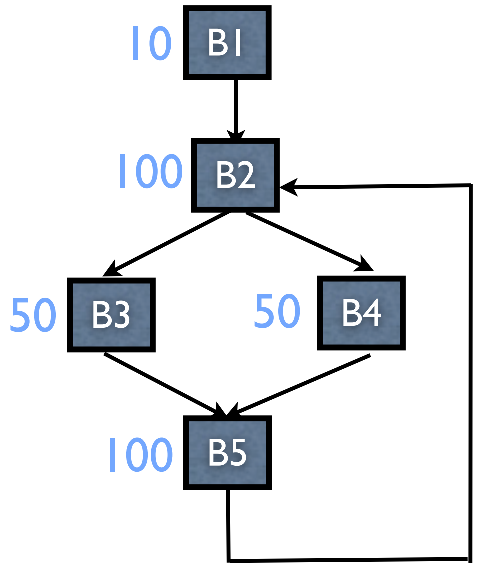
WILLIAM & MARY

Outline

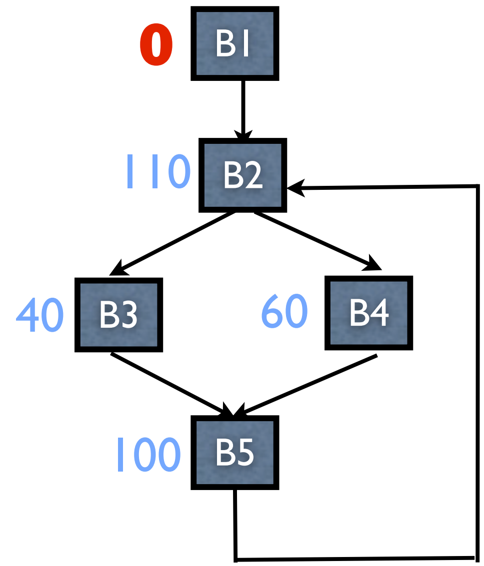
- Measuring the relations between sampling and FDO
- Analyzing critical sampling errors
  - Zero-counter error
  - Inconsistency error
- Statistical profile rectification

Zero-counter error

Exact profile:

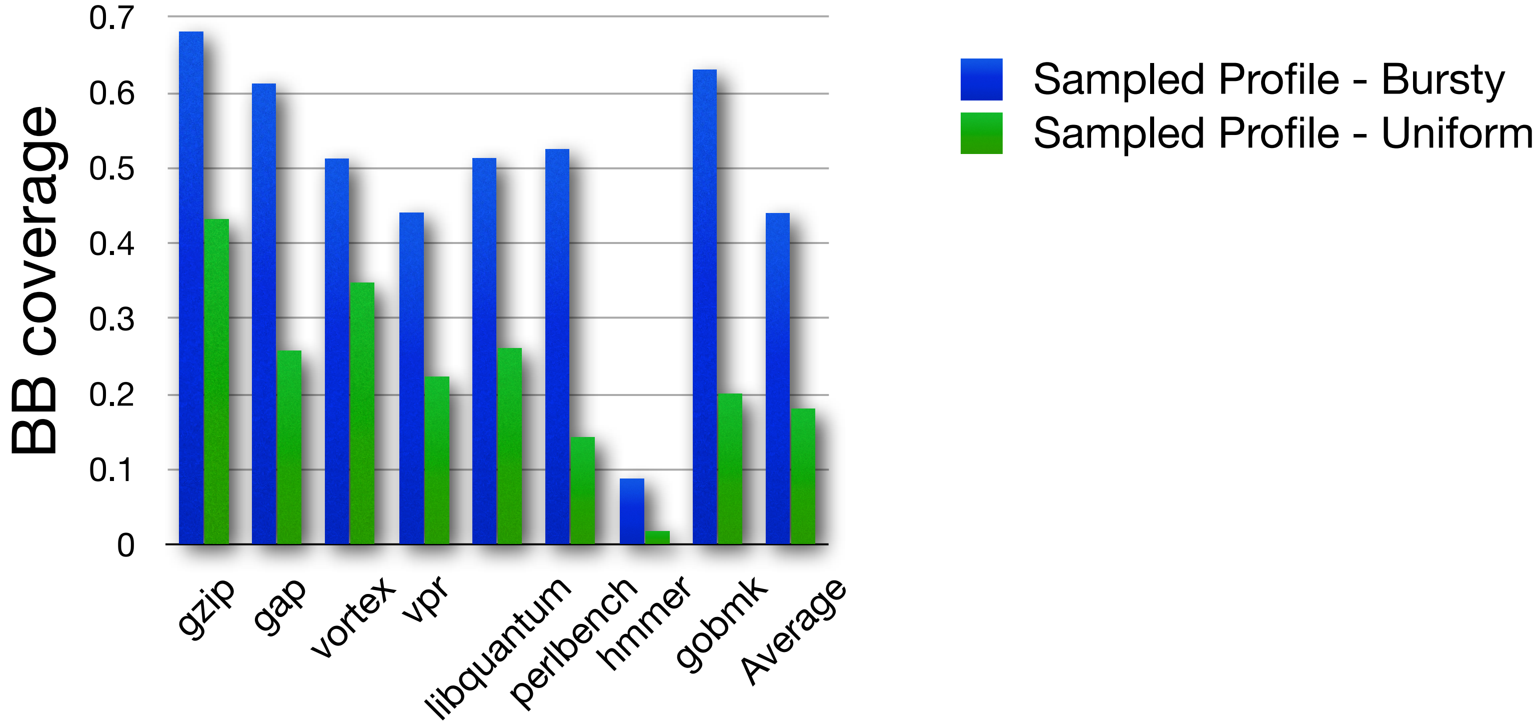


Scaled sampled profile:



Def: zero in the sampled profile, but not in the exact profile.
Shows the difference in coverage.

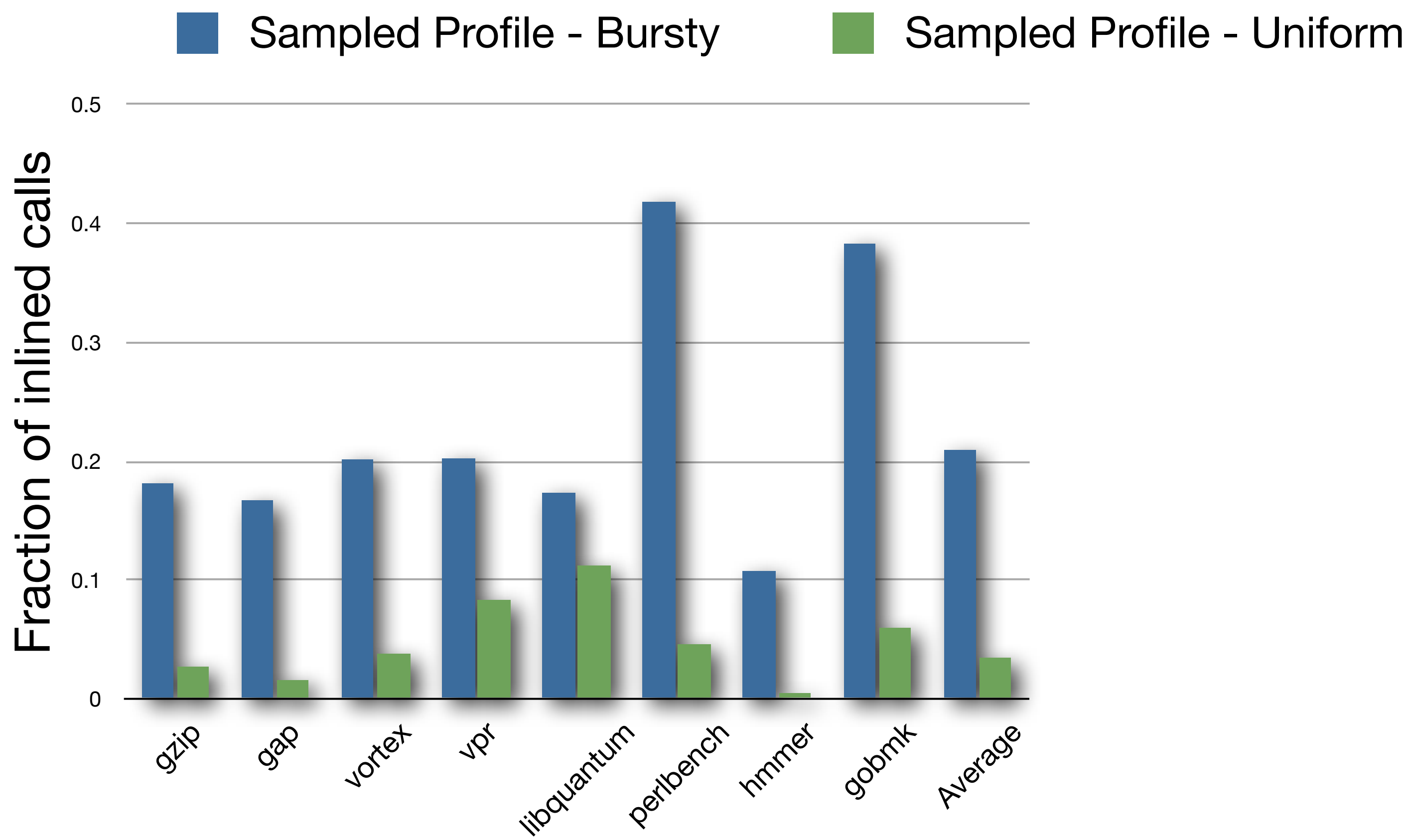
Zero-counter errors



- Many BBs are missed in a sampled profiles.

Influence on function inlining

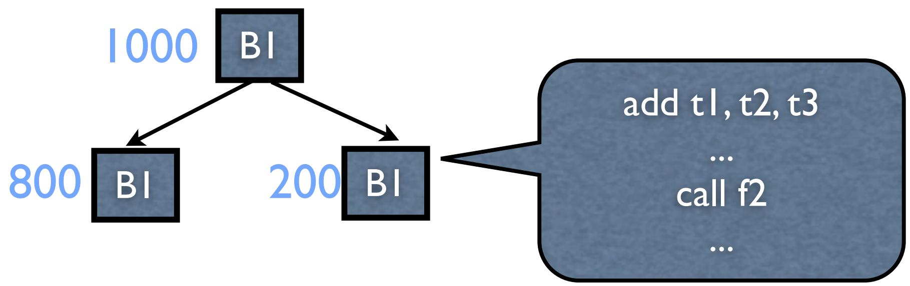
FDO on sampled profiles inlines much fewer calls.



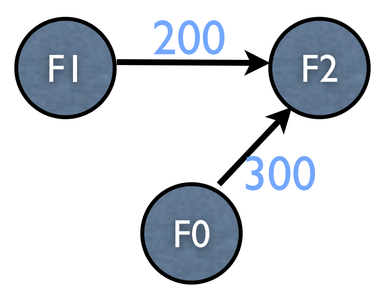
Influence on function inlining

- Different BB coverage ⇒ different dynamic call graphs

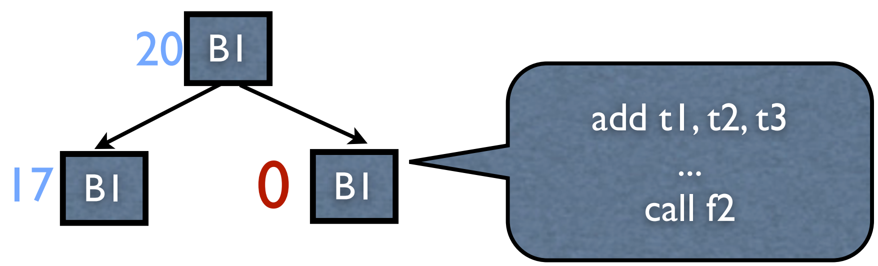
CFG in F1 (exact)



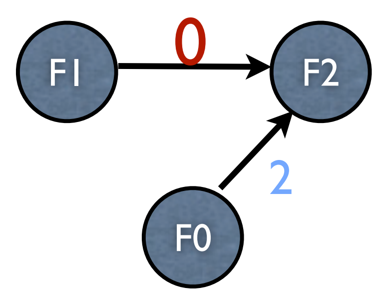
Dynamic call graph (exact)



CFG in F1 (sampled)

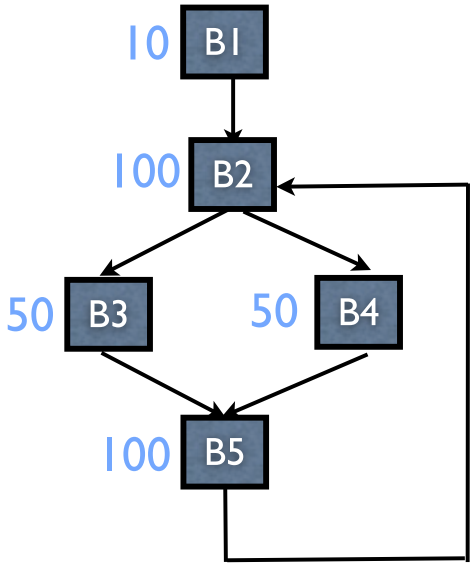


Dynamic call graph (sampled)

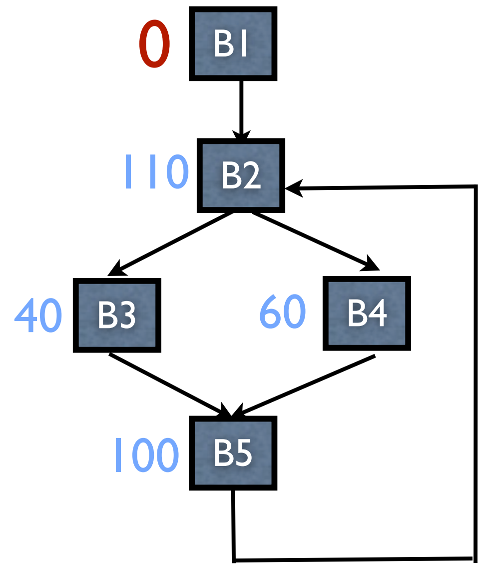


Influence on Loop Optimization

- Small counters are important for loop optimizations



Loop iteration: 100/10 = 10



Loop iteration: ?

- Influence on unrolling, specialization and many other transformations

WILLIAM & MARY

Outline

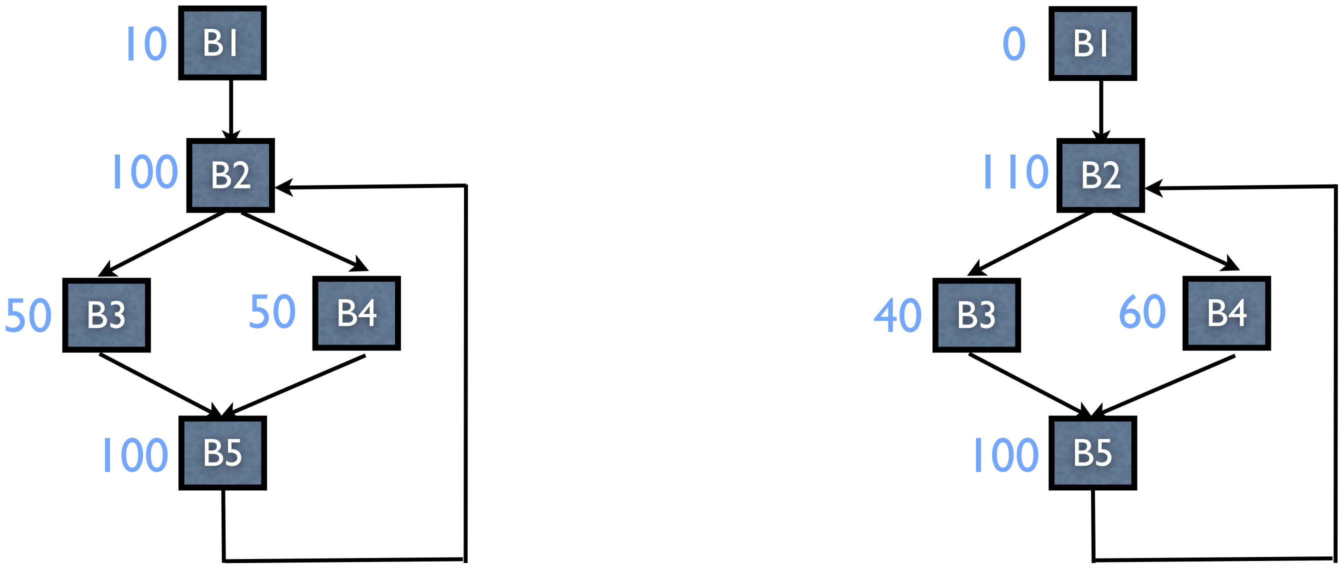
- Measuring the relations between sampling and FDO
- Analyzing critical sampling errors
  - Zero-counter error
  - inconsistency error
- Statistical profile rectification

WILLIAM & MARY

Inconsistency errors

1/10 sampling rate

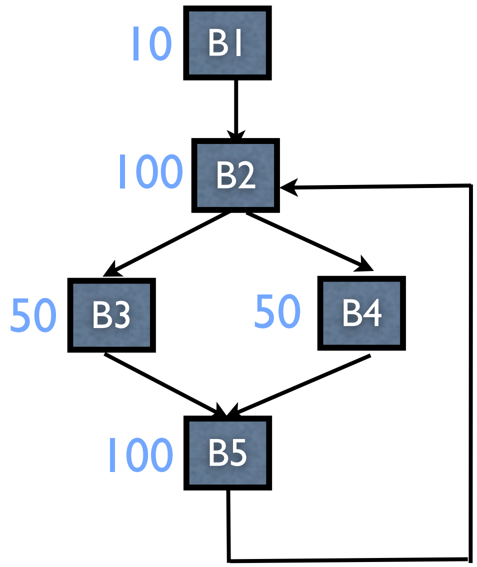
Exact BB counter profile: Scaled sample BB counter profile:



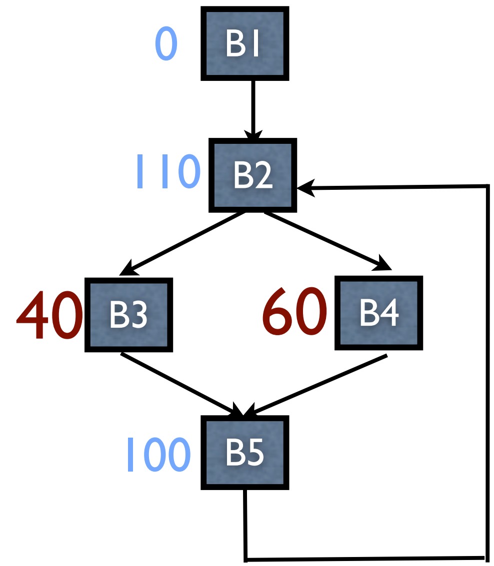
Inconsistency errors

1/10 sampling rate

Exact BB counter profile:



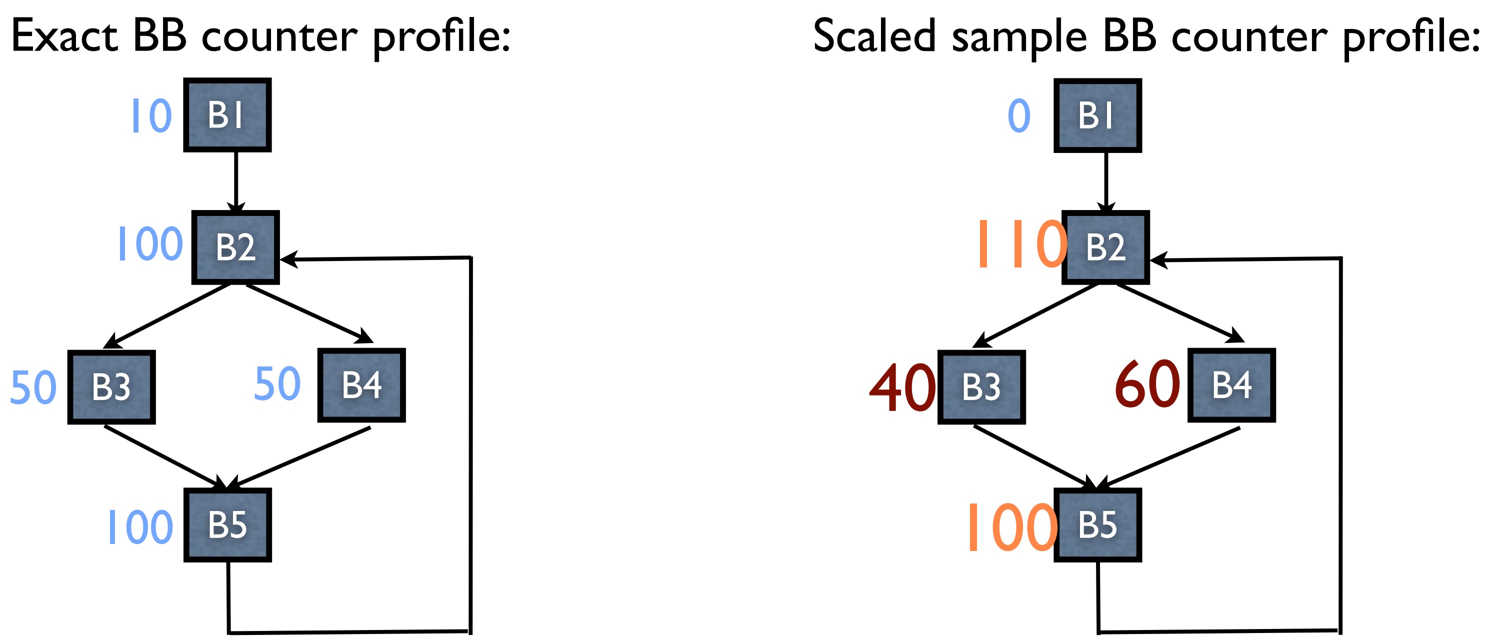
Scaled sample BB counter profile:



WILLIAM & MARY

Inconsistency errors

1/10 sampling rate

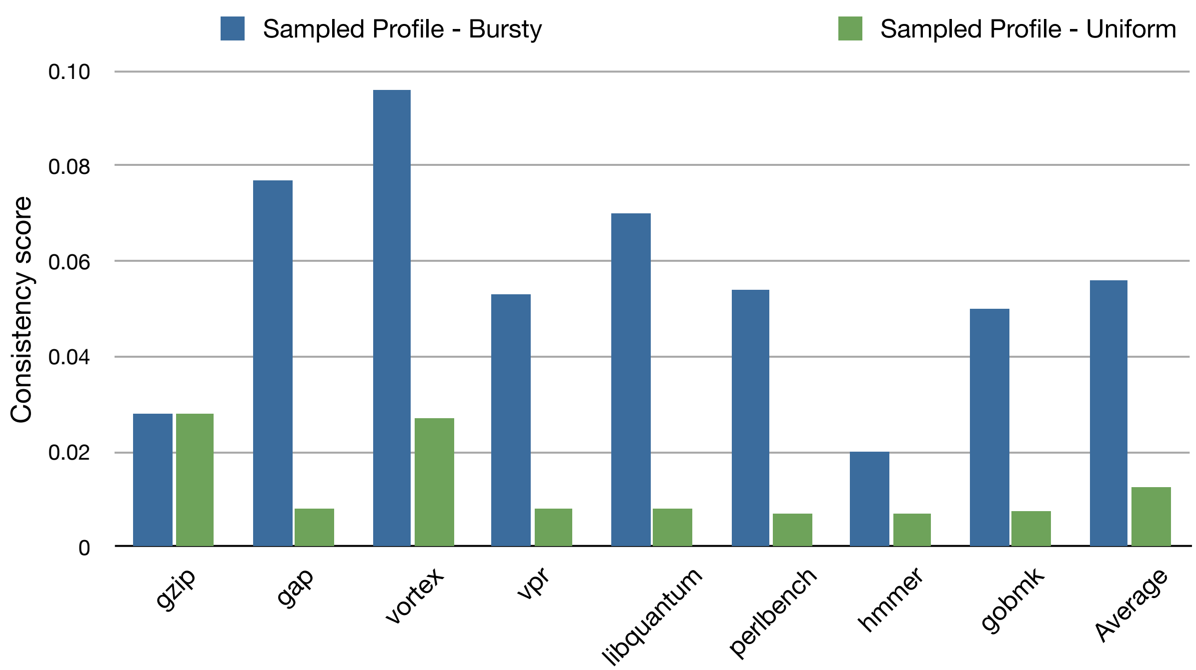


- Inconsistency between B2 and B5 fixable
- Hard to fix B3 and B4

Quantify the degree of inconsistency

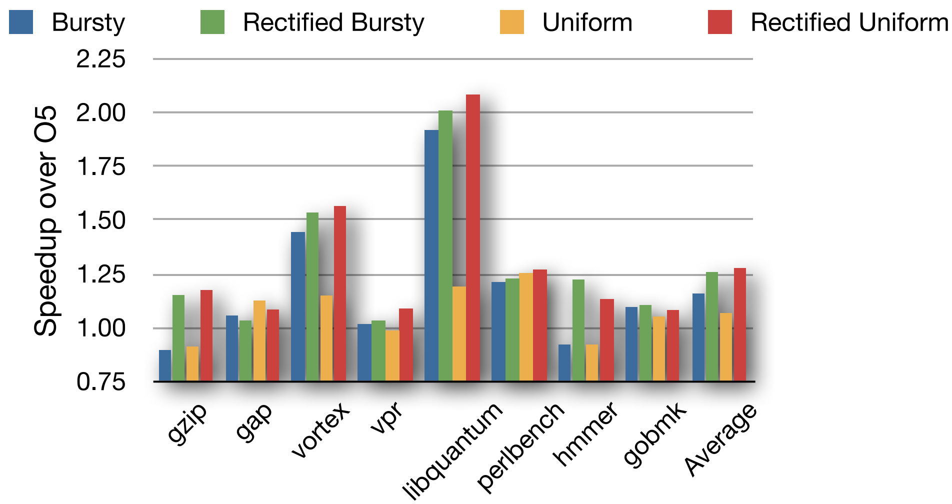
- G: a group of BBs having the same counter value
- G': the largest subset of G having the same counter value in the sample profile

Consistency Score(G) = |G'|/|G|

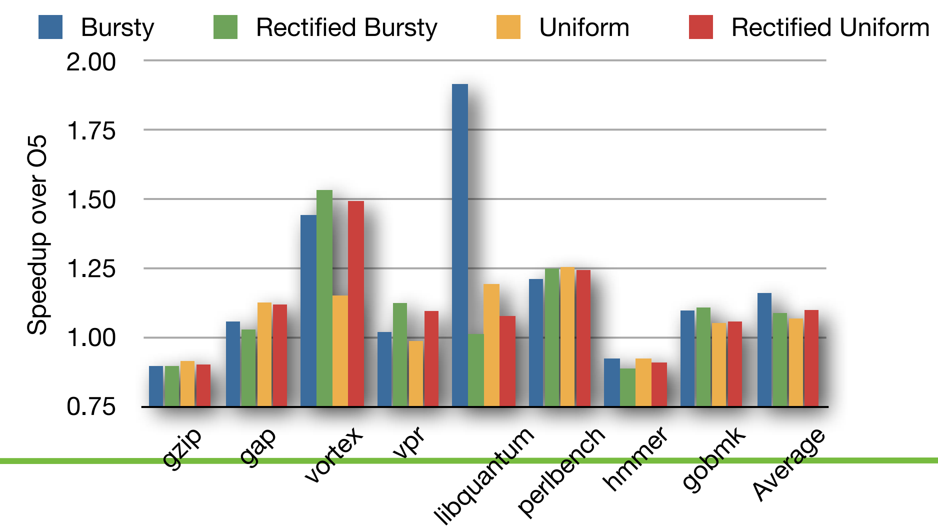


Potential of rectifying errors with exact profiles

Replace 0-counters with corresponding values in exact profiles



Replace inconsistent counters with their average



WILLIAM & MARY

Outline

- Measuring the relations between sampling and FDO
- Identifying critical sampling errors
- Statistical profile rectification

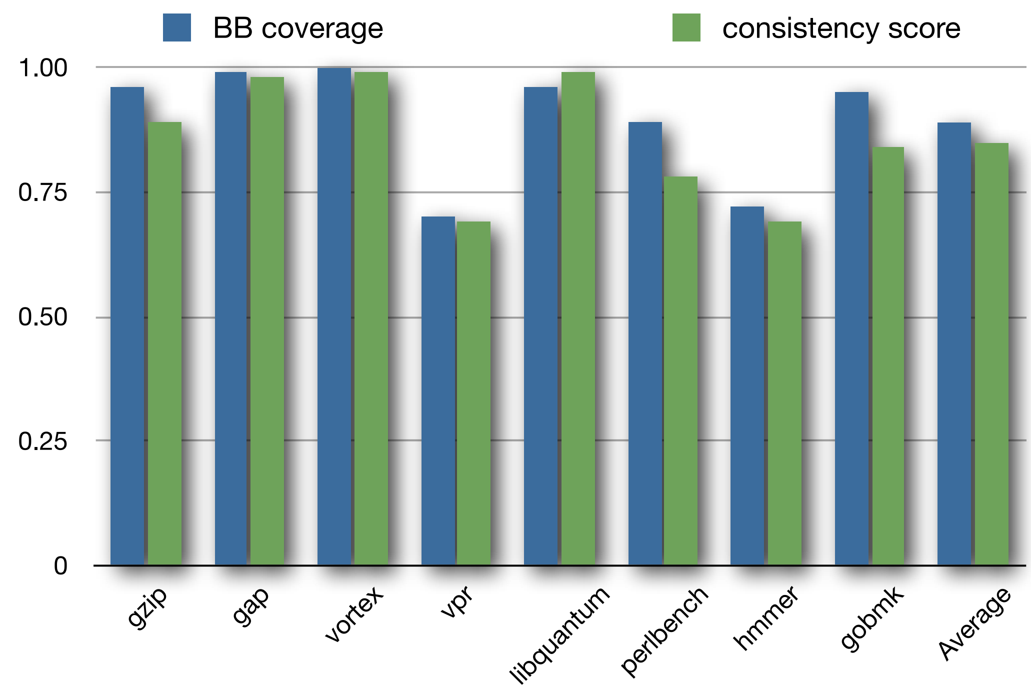
WILLIAM & MARY

Concept and options

- Profile rectification: Correct errors in a profile
- Option methods
  - Through static analysis
    - Conservative, subject to compile-time unknown
  - Through statistical patterns
    - Learn from training profiles

WILLIAM & MARY

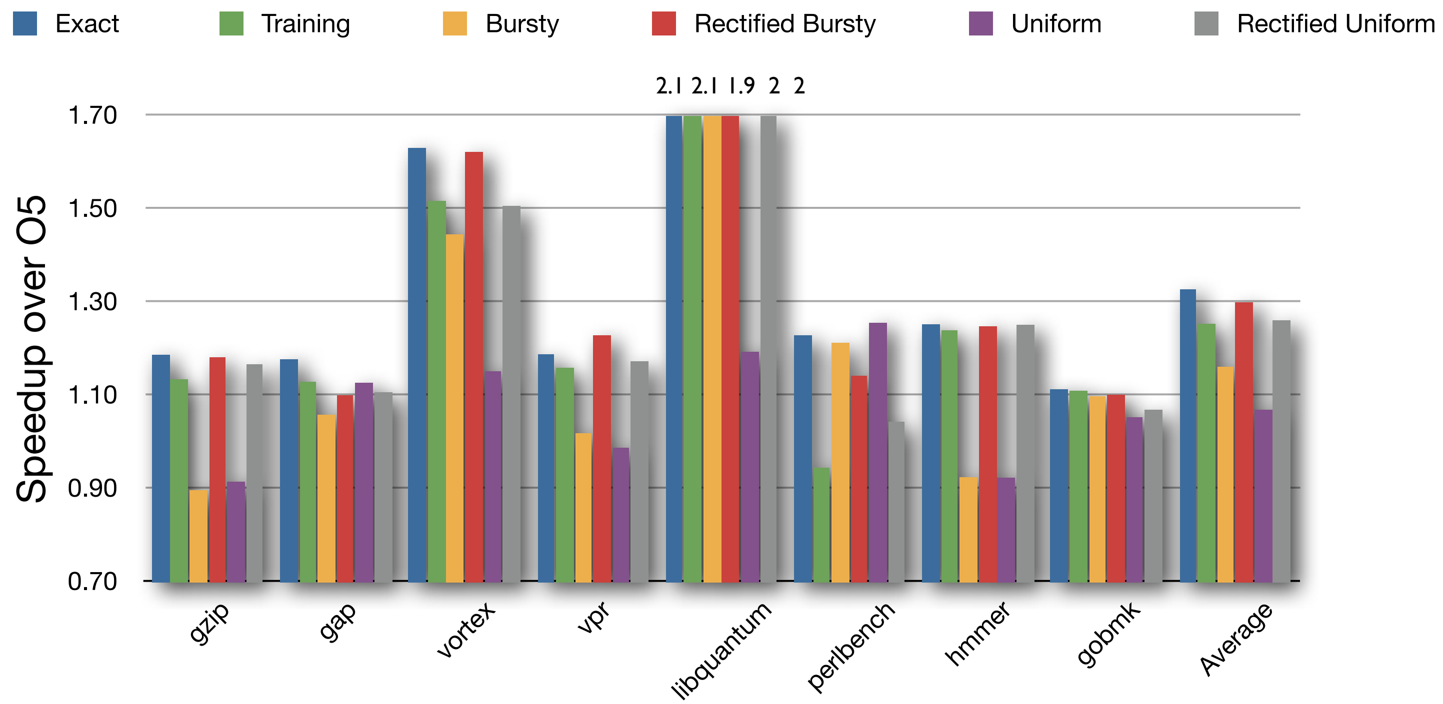
Cross-input stableness of patterns



- 3 ref inputs for each benchmark
- Average of pairwise comparison
- Suggest to use a training profiles to provide the pattern
- Note: pattern is stable, not the exact profile values

WILLIAM & MARY

Rectification using a training profile



- Both kinds of errors rectified
- Rectified profiles achieved around 90% benefit of FDO
- Rectified bursty profiles outperform training profiles

WILLIAM & MARY

Related Work

- Sample profile guided optimization in static compilers
  - ❖ Chen and others: Taming hardware samples [CGO'10]
- Profile errors
  - ❖ Mytkowicz and others: Java profile inconsistency [PLDI'10]
- Correlation between profile accuracy and usefulness
  - ❖ Langdale and Gross [DDD'03]

WILLIAM & MARY

Conclusion

- Counter-intuitive findings on sampling for FDO
  - ❖ intuition: higher sampling rates -> more accurate profiles
    - ❖ reality: strong correlations for bursty sampling, but not for uniform sampling
  - ❖ intuition: more accurate profiles -> better performance
    - ❖ reality: weak correlations
- Two types of critical errors in sampled profiles
  - ❖ 0-counter errors & inconsistency errors
- Effectiveness of simple statistical profile rectification
  - ❖ 92% of the speedups that exact profiles provide

WILLIAM & MARY