



# Correcting the Dynamic Call Graph Using Control Flow Constraints

**Byeongcheol (BK) Lee**  
Kevin Resnick  
Michael Bond  
Kathryn McKinley  
UT Austin

To appear in CC 2007

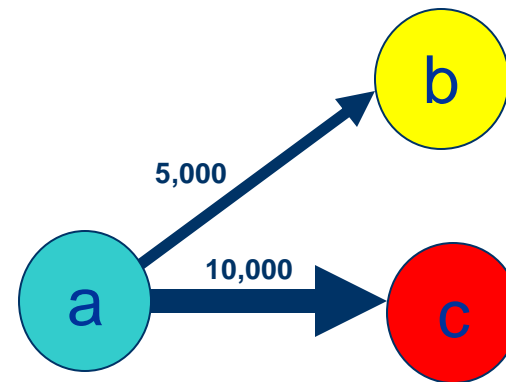
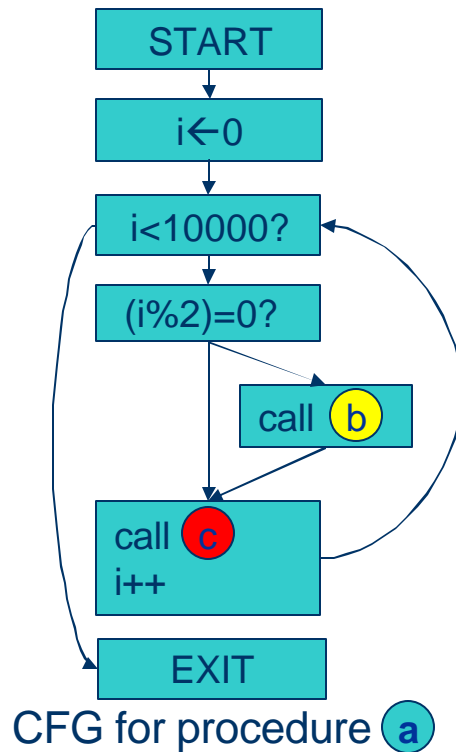


# Motivation

- Object oriented programs are becoming large and complex
- Dynamic call graph (DCG)
  - Presents the whole program behavior
  - Shows how frequently each method call is activated
  - Critical input to the inter-procedural optimizations
- High accuracy call graph is desired

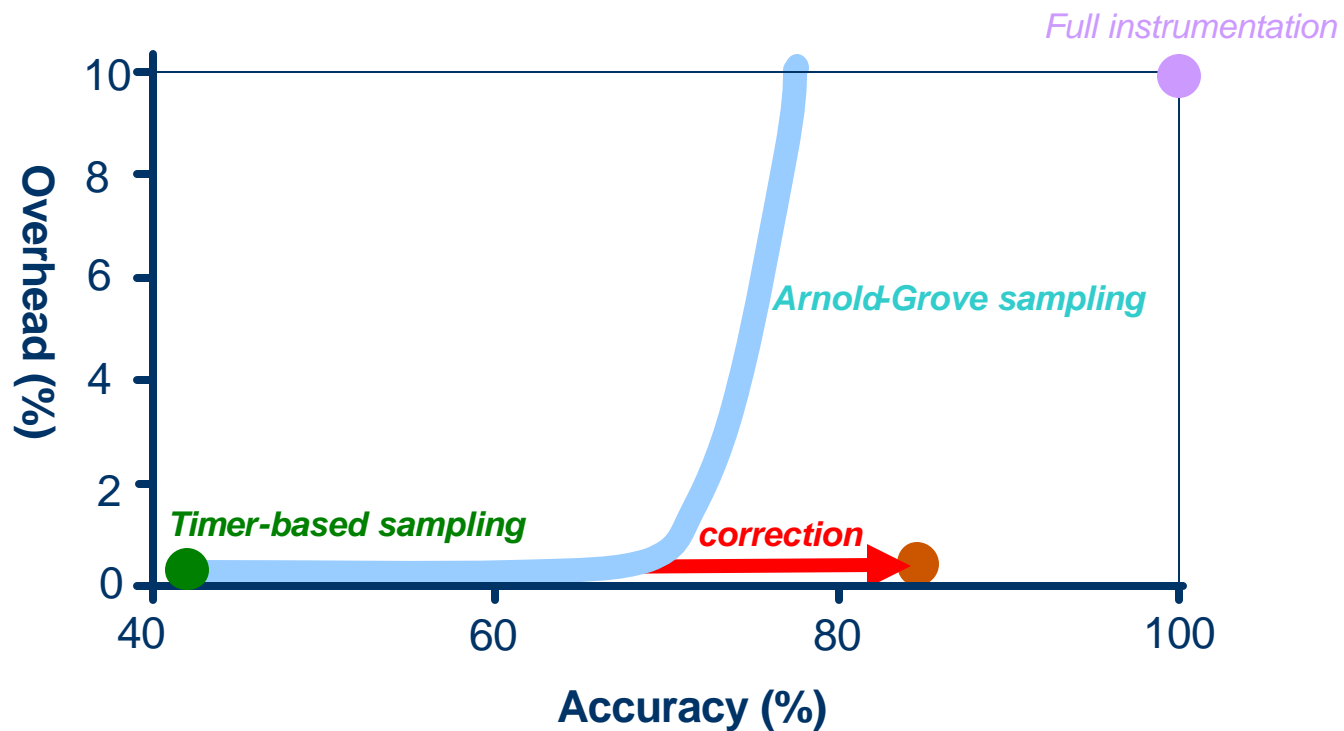


# Control Flow Graph (CFG) and Dynamic Call Graph (DCG)



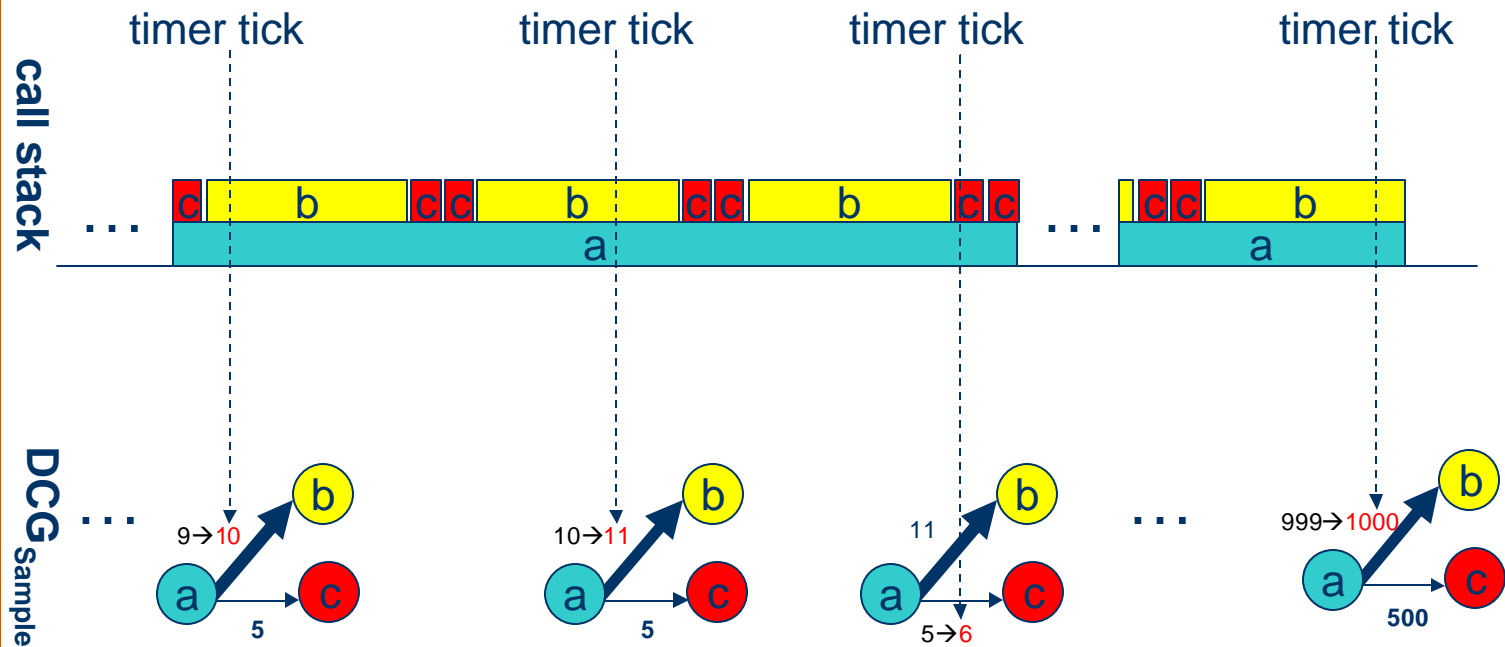


# Call graph profiling



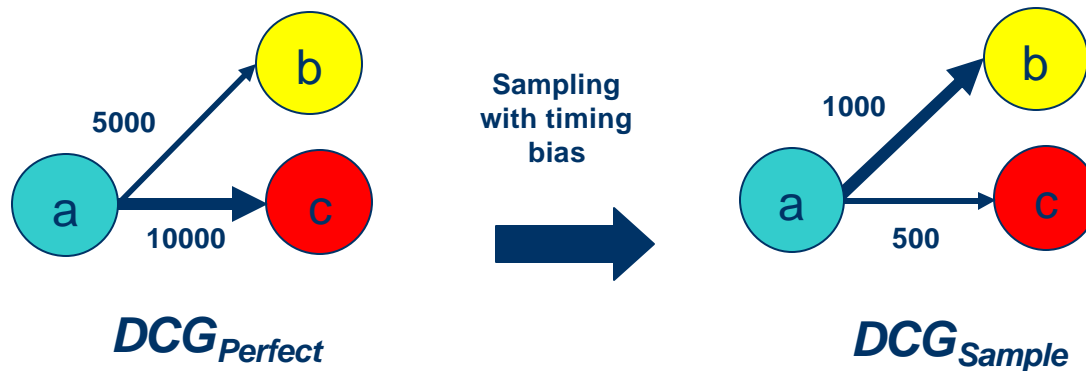


# Sampling and timing Bias





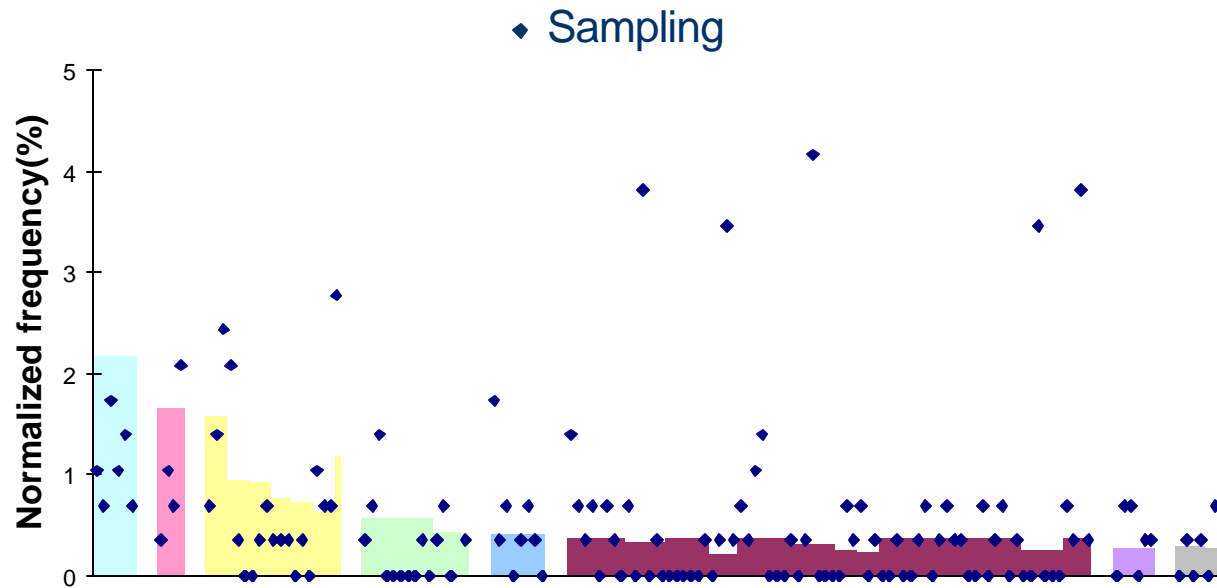
# Timing bias misleads optimizer



- $DCG_{Sample}$ 
  - Edge frequencies were reversed!
- Inlining decision
  - Inliner may inline **b** instead of **c**



# Timing bias in SPECJVM98 *raytrace*






# Outline

- Motivation
- Call graph correction
- Accuracy & performance



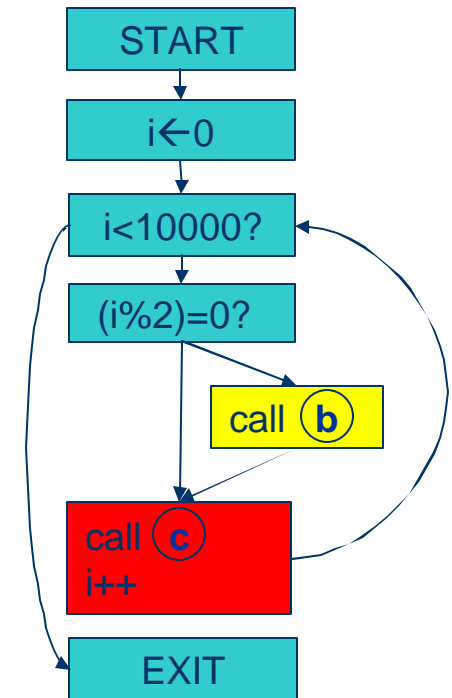
# Correction algorithms

- Modify DCG to satisfy constraints from CFG
  - CFG restricts possible DCG
- Static and dynamic approaches
  - Static FDOM (Frequency dominator) correction
    - Static approach 
    - Uses static FDOM constraint on DCG
  - Dynamic basic block profile correction
    - Dynamic approach
    - Uses dynamic basic block profile constraint on DCG



# Static FDOM constraint

- FDOM constraint on CFG
  - **call c** is executed at least as many times as **call b**
  - **call c** FDOM **call b**
- FDOM constraint on DCG
  - $f(a \rightarrow c) = f(a \rightarrow b)$

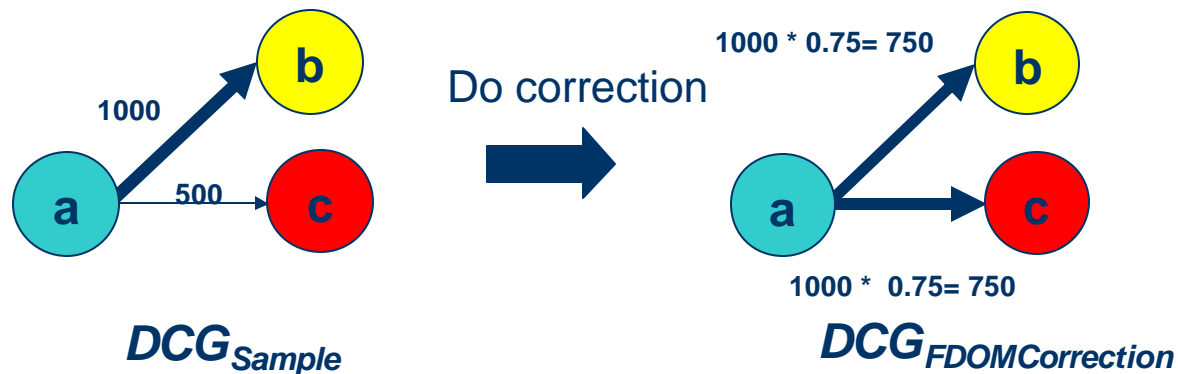


CFG for procedure **a**



# Static FDOM correction

FDOM Constraint:  $f(a \rightarrow c) = f(a \rightarrow b)$

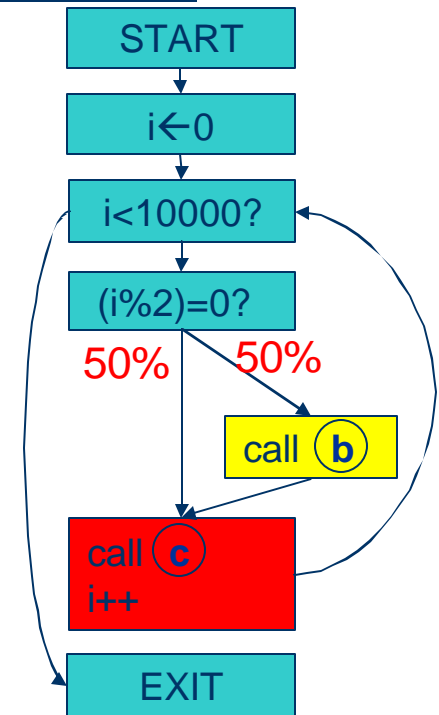


Assign the max frequency to satisfy FDOM constraint



## Dynamic basic block profile constraint

- Some dynamic optimization systems do edge profiling
  - Baseline compiler in Jikes RVM
- Dynamic basic block profile constraint on CFG
  - $f(\text{call } \textcircled{c}) = 2 * f(\text{call } \textcircled{b})$
- Dynamic basic block profile constraint on DCG
  - $f(\textcircled{a} \rightarrow \textcircled{c}) = 2 * f(\textcircled{a} \rightarrow \textcircled{b})$

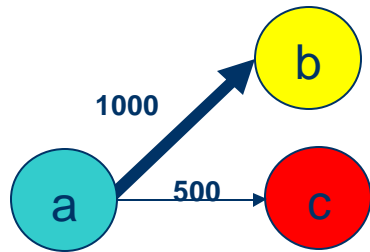


CFG for procedure  $\textcircled{a}$



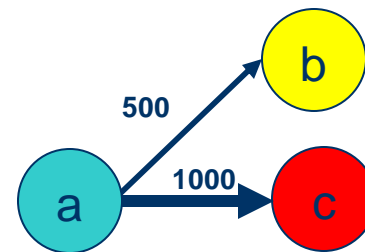
# Dynamic basic block profile correction

Dynamic basic block profile constraint:  $f(a \rightarrow c) = 2 * f(a \rightarrow b)$



*DCG*<sub>Sample</sub>

Do correction



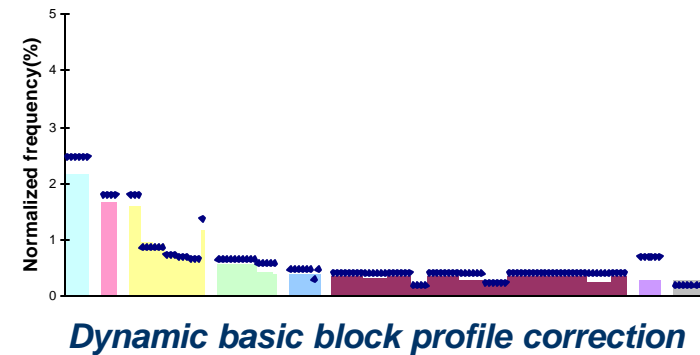
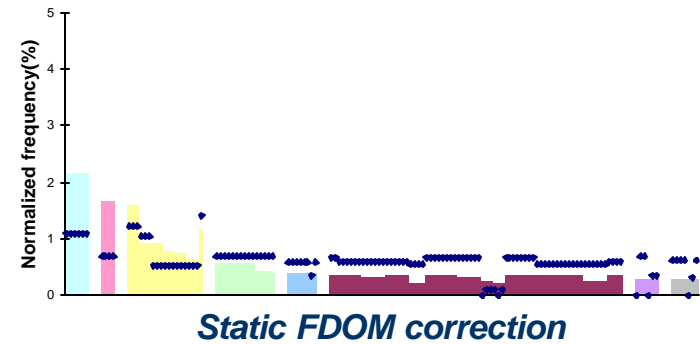
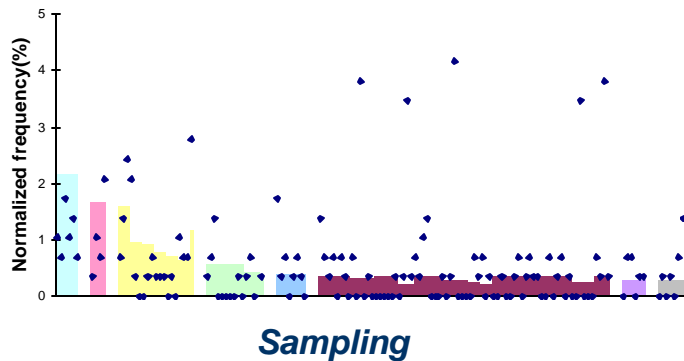
*DCG*<sub>EdgeProfileCorrection</sub>

$$f_{\text{New}}(a \rightarrow b) = 1/(1+2) * (1000+500) = 500$$

$$f_{\text{New}}(a \rightarrow c) = 2/(1+2) * (1000+500) = 1000$$



# Call graph correction for *raytrace*





# Outline

- Motivation
- Call graph correction
- Accuracy & performance

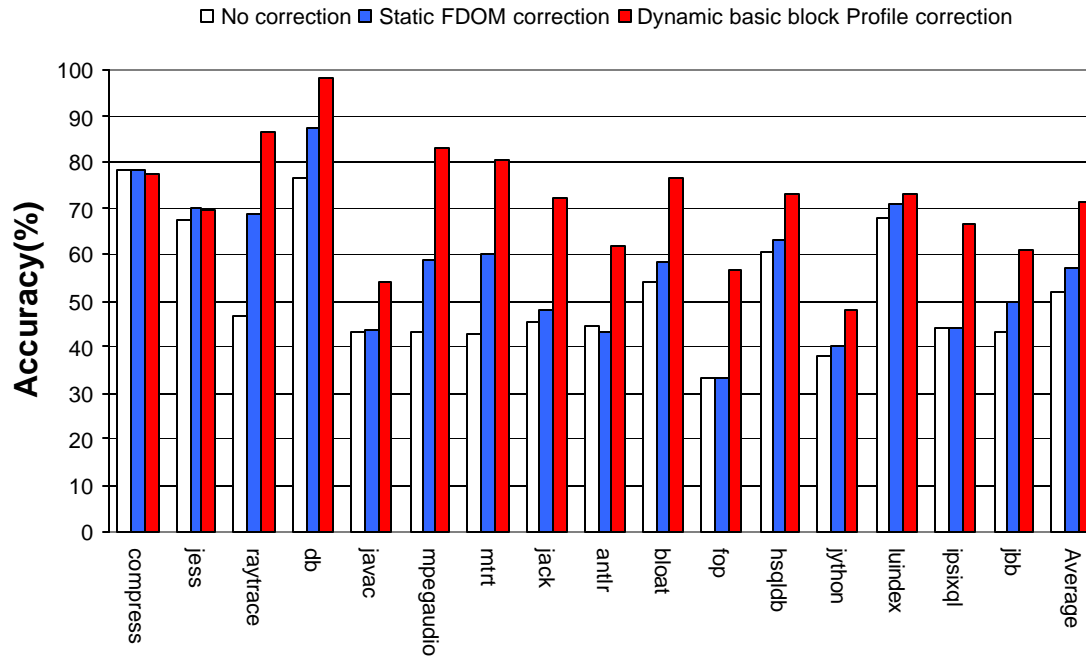


# Experimental methodology

- Jikes RVM 2.4.5 on 3.2G Pentium 4
- Two metrics for evaluation
  - Call graph accuracy
    - Use overlap accuracy [Arnold & Grove '05]
  - Execution time
    - Replay methodology to control nondeterminism [Steve et al. '06]
- SPECJVM98 and DaCapo benchmarks

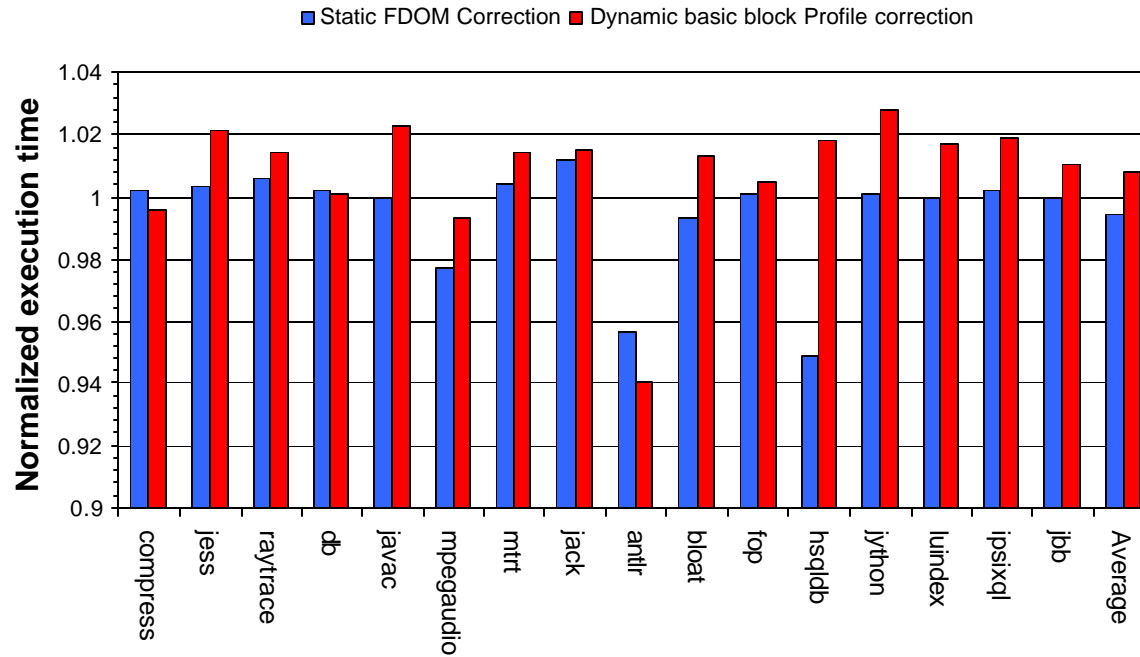


# Accuracy



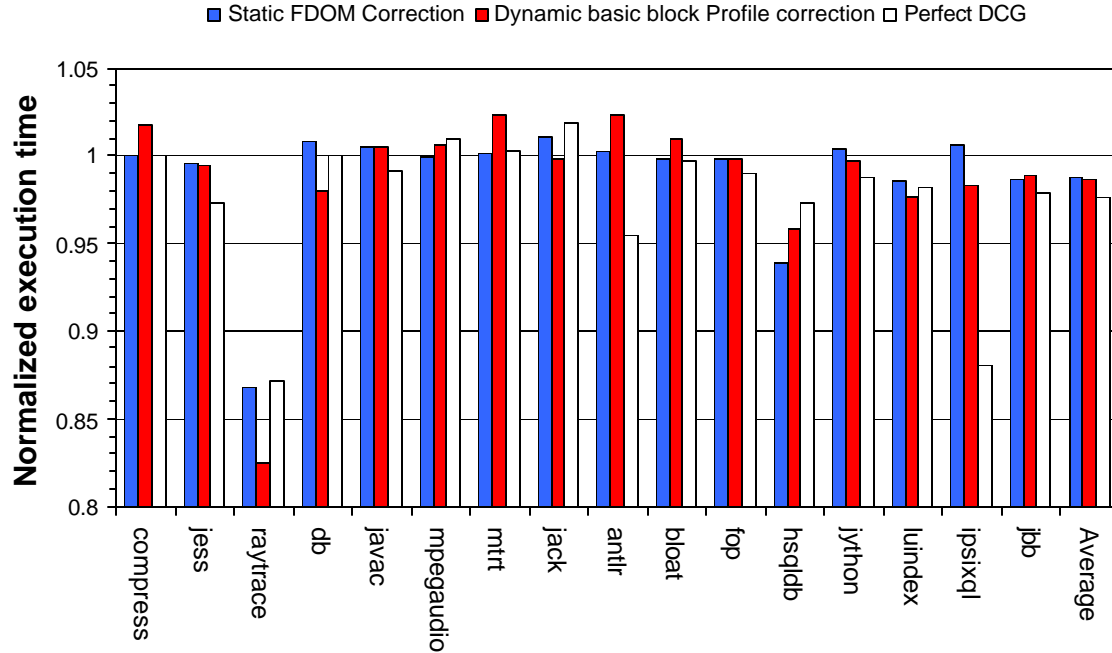


# Overhead





# Inlining performance





# Summary

- CFG constraint improves the DCG
- Advantages
  - Can be easily combined with other DCG profiling
  - Minimal overhead only during the compilation
- Future work
  - More inter-procedural optimizations with high accuracy DCG