



Euro LLVM  
London, England

April 13-14, 2015

# LLVM Inliner Enhancement

Jiangning Liu  
Kevin Qin



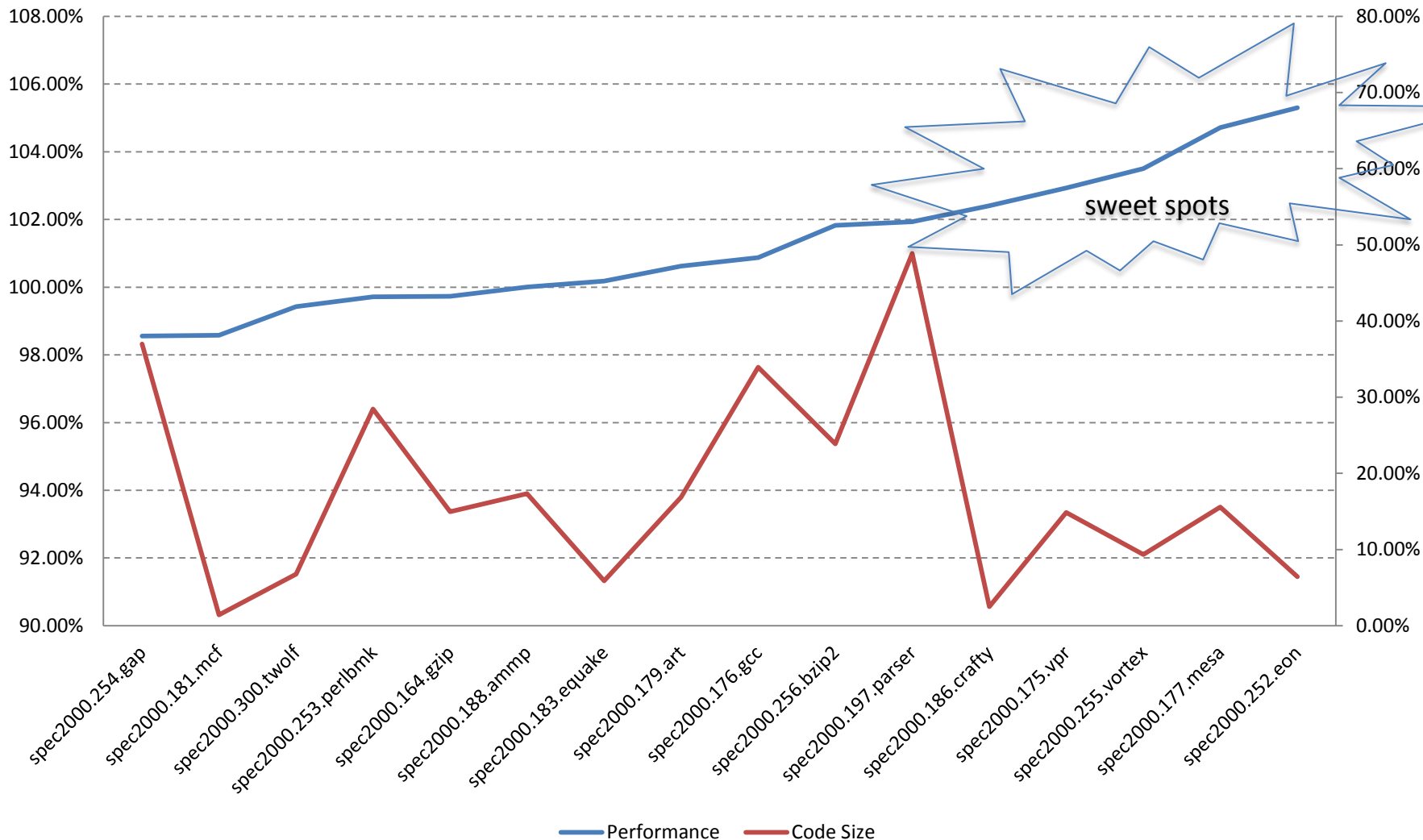


# Background

- Carefully tuned for a large scope of real applications.
- Some missing opportunities for typical computation intensive benchmarks.



## Performance Gain and Code Size Bloat for SPEC2000 After Using `-inline-threshold=1000`



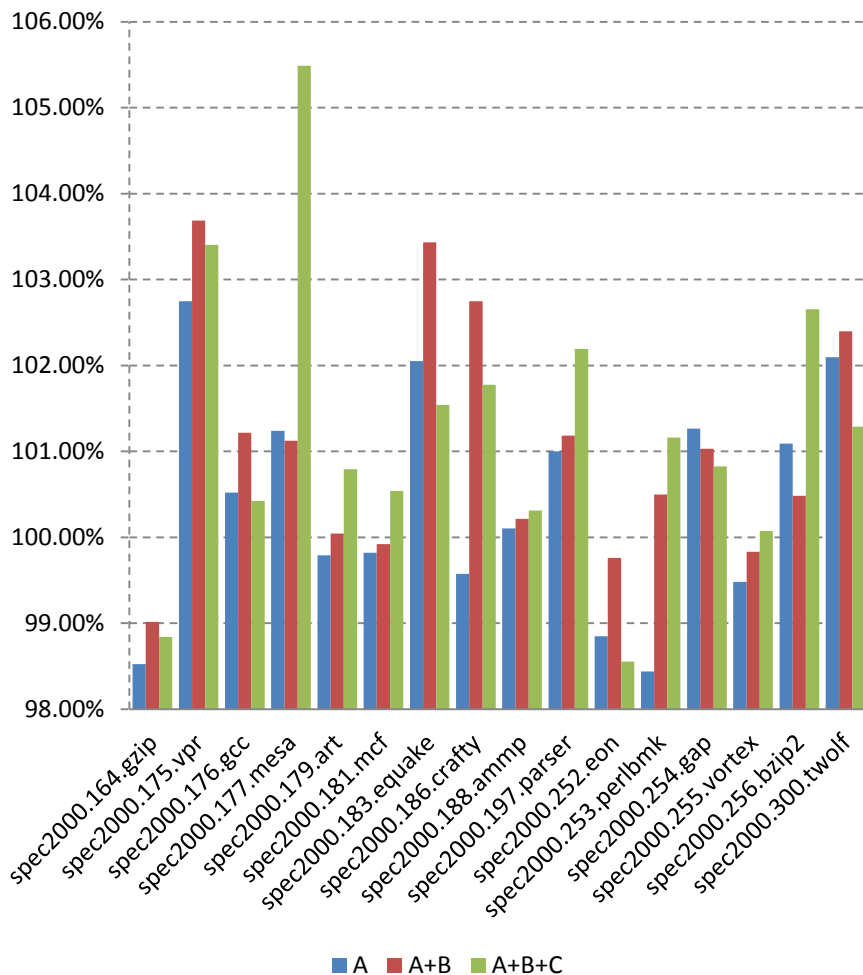


# Heuristic Rules

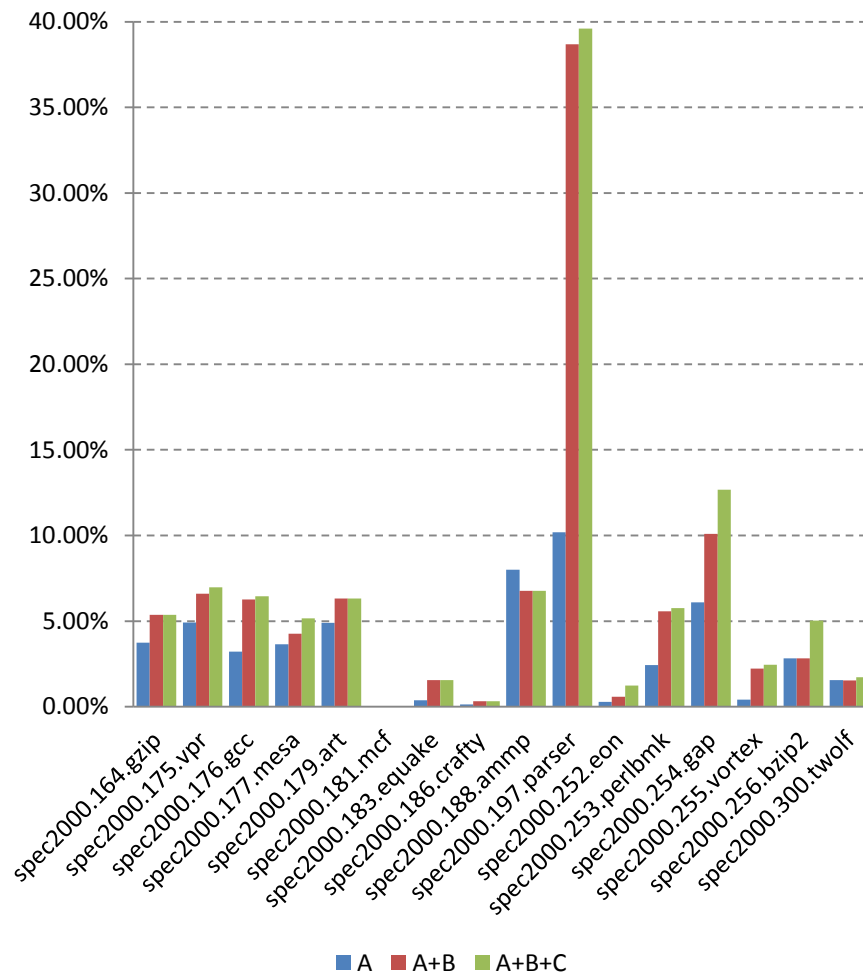
- Performance
  - (A) 2X threshold for callee inside loop.
  - (B) 2X threshold for callee with const argument.
  - (C) 4X threshold for callee inside loop with  $\leq 3$ BB.
- Code Size
  - (D) Reduce threshold to 225 for \*cold\* callees.



## SPEC2000/2006 Performance Change on AArch64 (r226173)

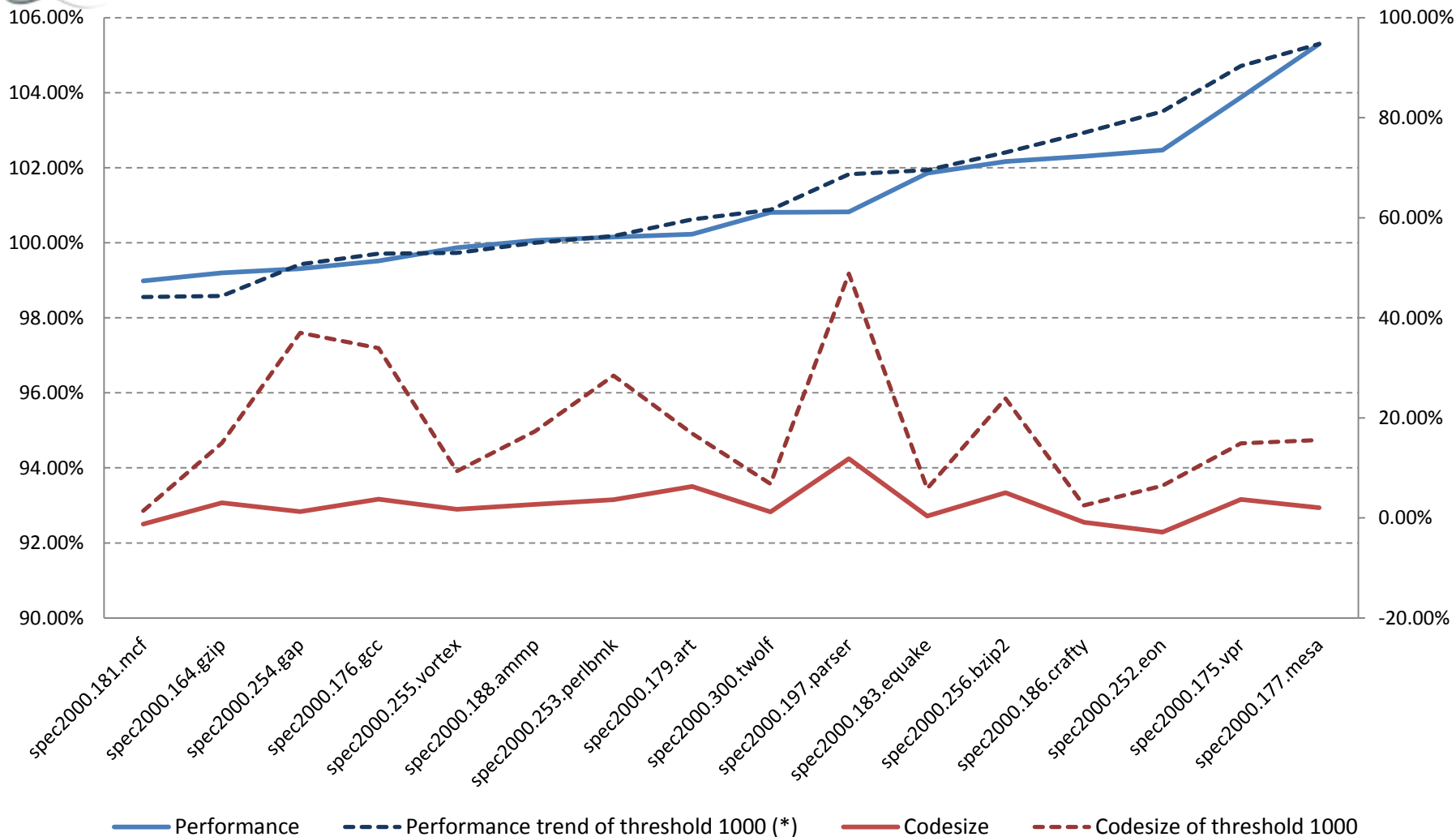


## SPEC2000/2006 Code Size Change for AArch64 (r226173)





## SPEC2000 Performance Gain and Code Size Bloat after applying heuristic rule A+B+C+D (r226173)

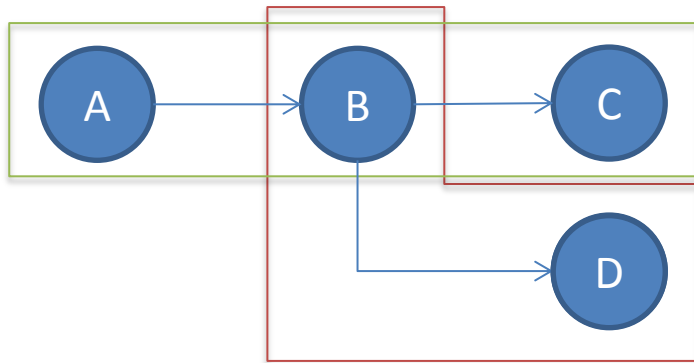


\* Show trend only! The x axis positions don't 1:1 match with the names in this chart!

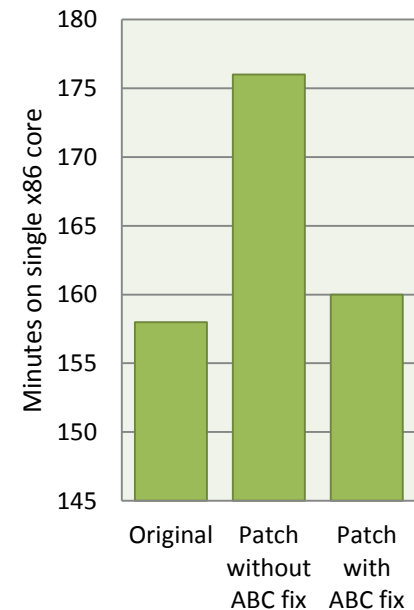


# Compile Time

- Loop Info Analysis is expensive
- Fix A->B->C issue
  - Early exit
  - Choose A->B->C rather than B->D

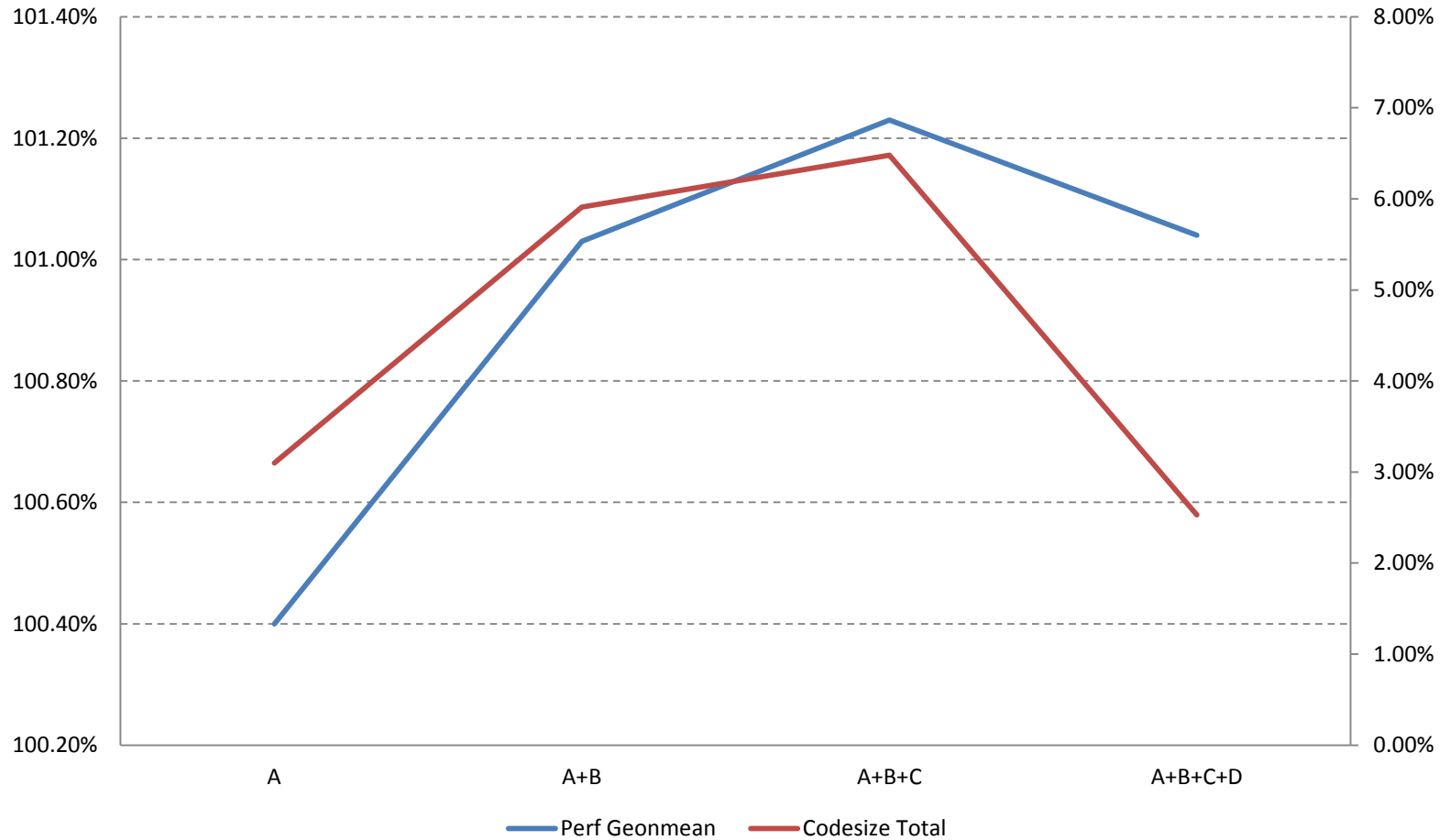


**llvm bootstrap time  
(r226173)**





## Trade-off between performance and codesize for SPEC2000 (r226173)







# Current Status

- Patch is under review in community.



Thank you!



# Challenges

- Trade-off
  - Performance gain
  - Code size bloat
  - Compile-time slowdown



# SPEC2006 Performance Gain and Code Size Bloat after applying heuristic rule A+B+C+D (r226173)

