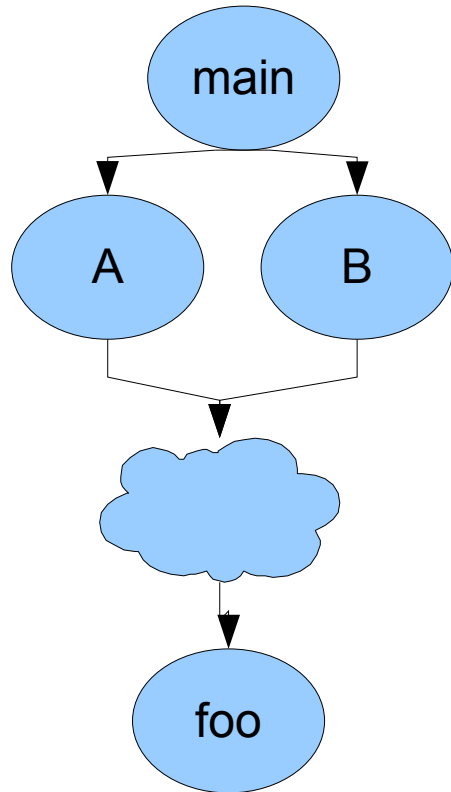


Inferred Call Path Profiling (ICPP)

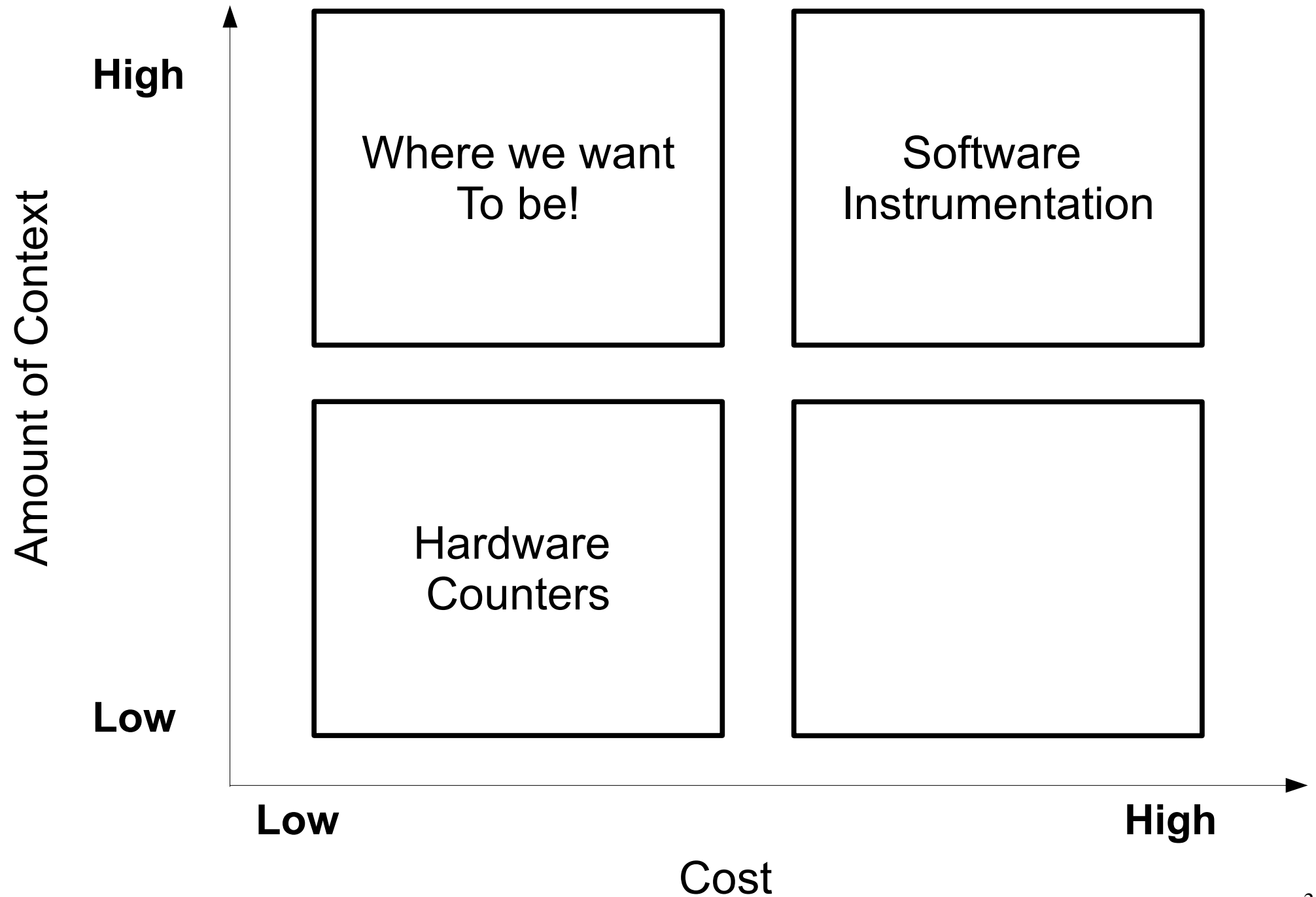
Todd Mytkowicz, Devin Coughlin and Amer Diwan
University of Colorado at Boulder

Call Paths vs Calling Context

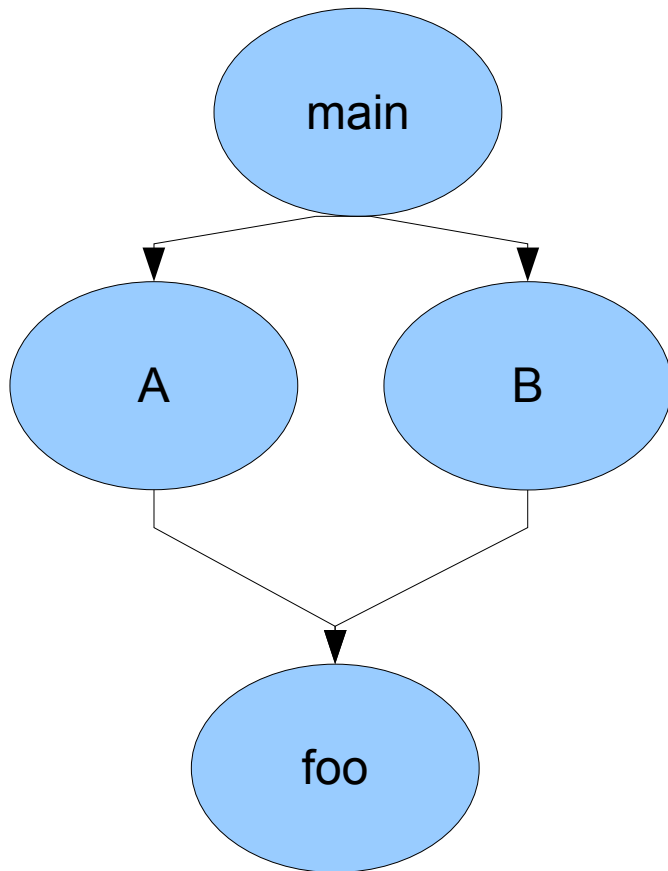


Call Paths: Sequence of calls
 $\text{main} \rightarrow A \rightarrow \dots \rightarrow \text{foo}$

Calling Context: Value that represents
sequence of calls

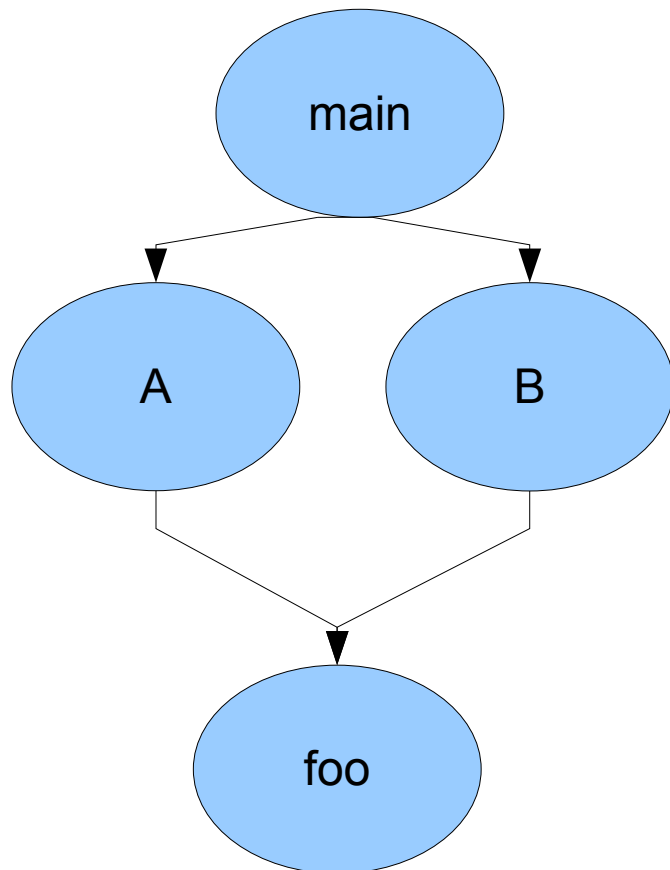


Bond and McKinley's Probabilistic Calling Context



```
void A() {  
    ...  
    foo();  
    ...  
}
```

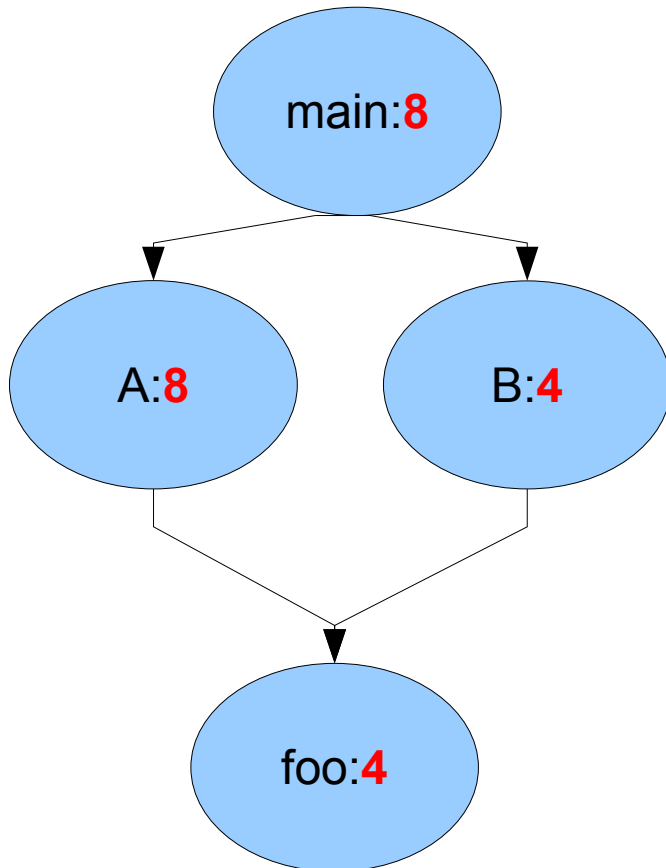
Bond and McKinley's Probabilistic Calling Context



```
void A() {  
    int temp = V;  
  
    ...  
    V = f(temp, cs_1);  
    cs_1: foo();  
    ...  
}
```

V at foo provides calling context

Context *without* computation?



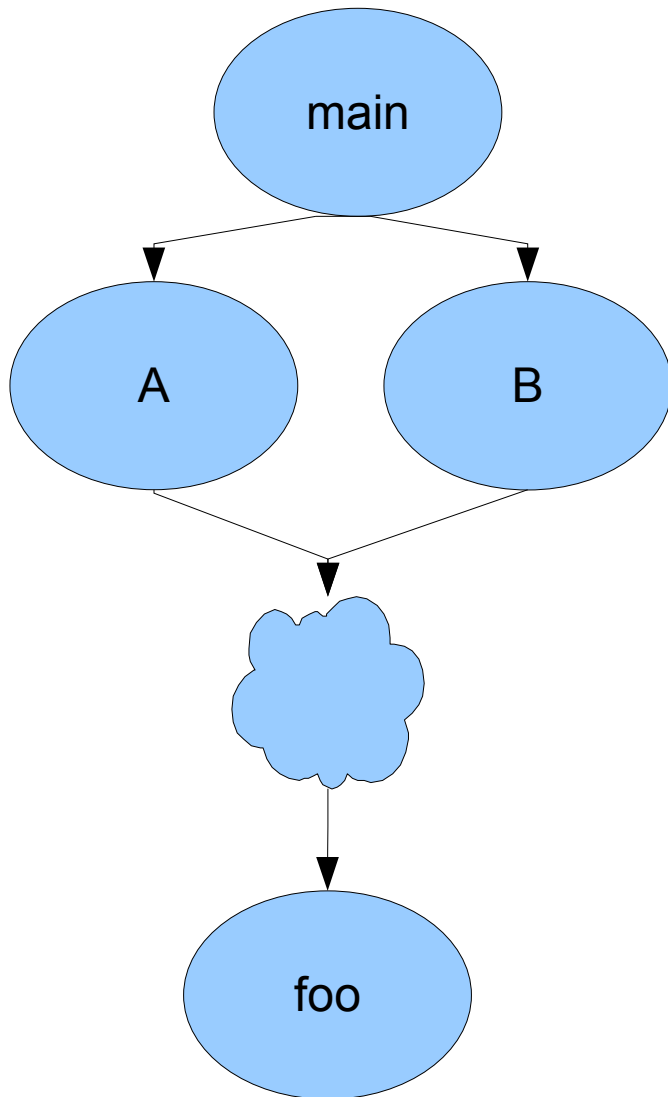
Stack Height: Number of bytes from main to foo

foo with stack height of 20 bytes:
main → A → foo

foo with stack height of 16 bytes:
main → B → foo

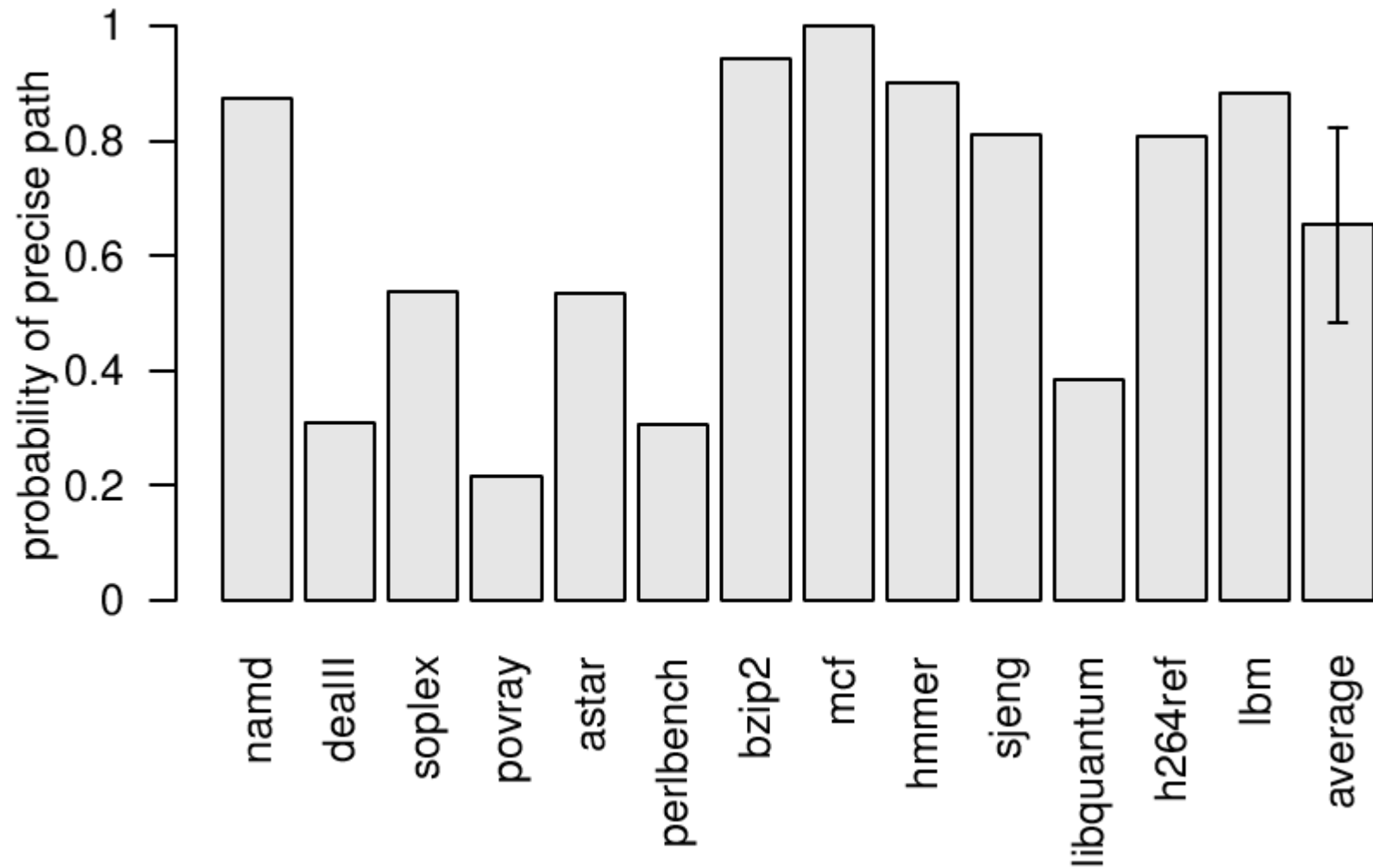
SP at foo provides calling context

Inferring call paths



program-counter = foo
stack height = 48 bytes

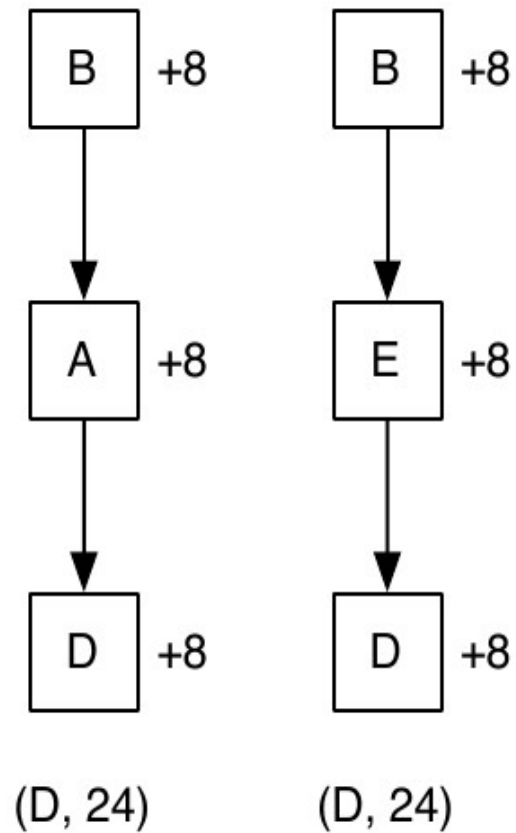
What is the call path?



PC/SP is precise 66% of the time

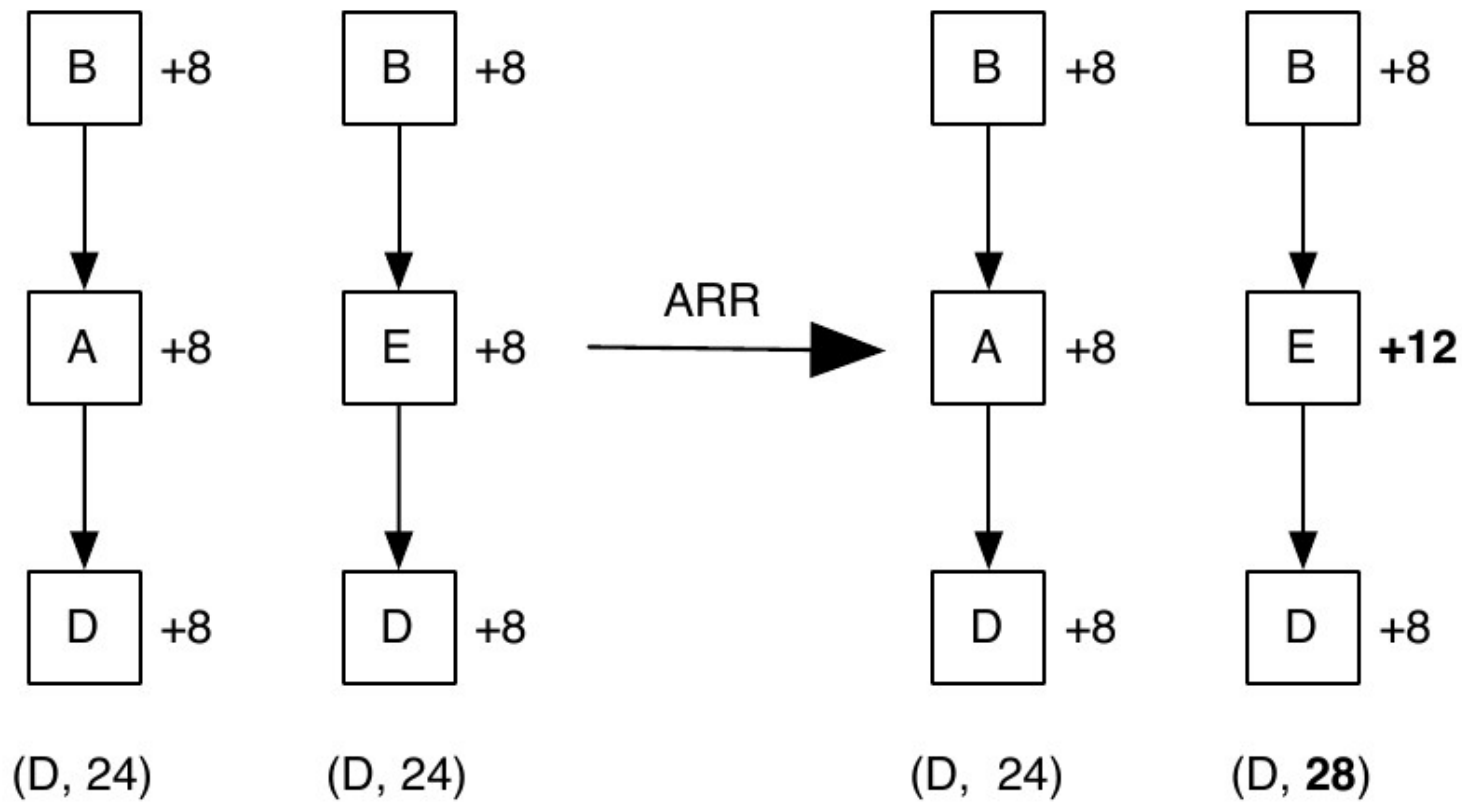
Problem: *Ambiguity*

- PC/SP maps to more than one call path



Remove ambiguity by changing height of one call path

Activation Record Resizing



ARR via the Frame Pointer

Before ARR:

```
<Perl_av_store>:  
402de8:  push %rbp  
402de9:  mov  %rsp, %rbp  
402dec:  sub  $0x30, %rsp
```

After ARR:

```
<Perl_av_store>:  
402de8:  push %rbp  
402de9:  mov  %rsp, %rbp  
402dec:  sub  $0x40, %rsp
```

Disambiguation: Random Search

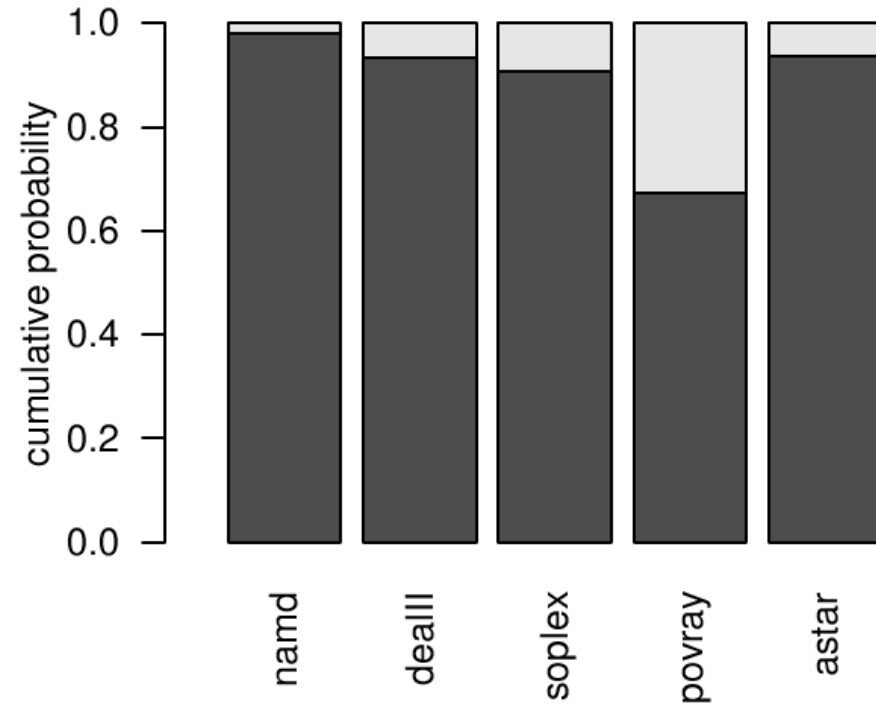
- Produce PC/SP to call path map
- Find random conflict
- Affect height of one path
 - accept if % ambiguity goes down
- Repeat until done

Experimental Methodology

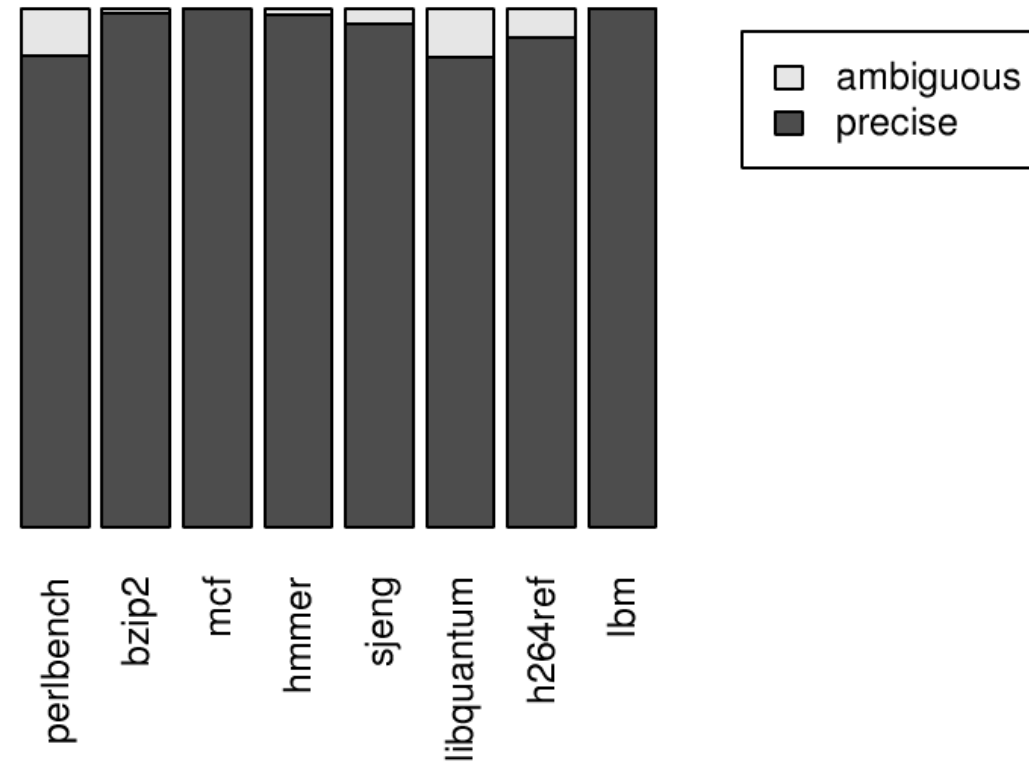
- Intel ICC compiler on 2.4GHz workstation
- SPEC C/C++ CPU 2006
- Offline: train and evaluate on *same* data
 - SPEC train input set
- Online: train and evaluate on *different* data
 - SPEC train and ref input sets

Results: Offline Scenario

SPEC C++



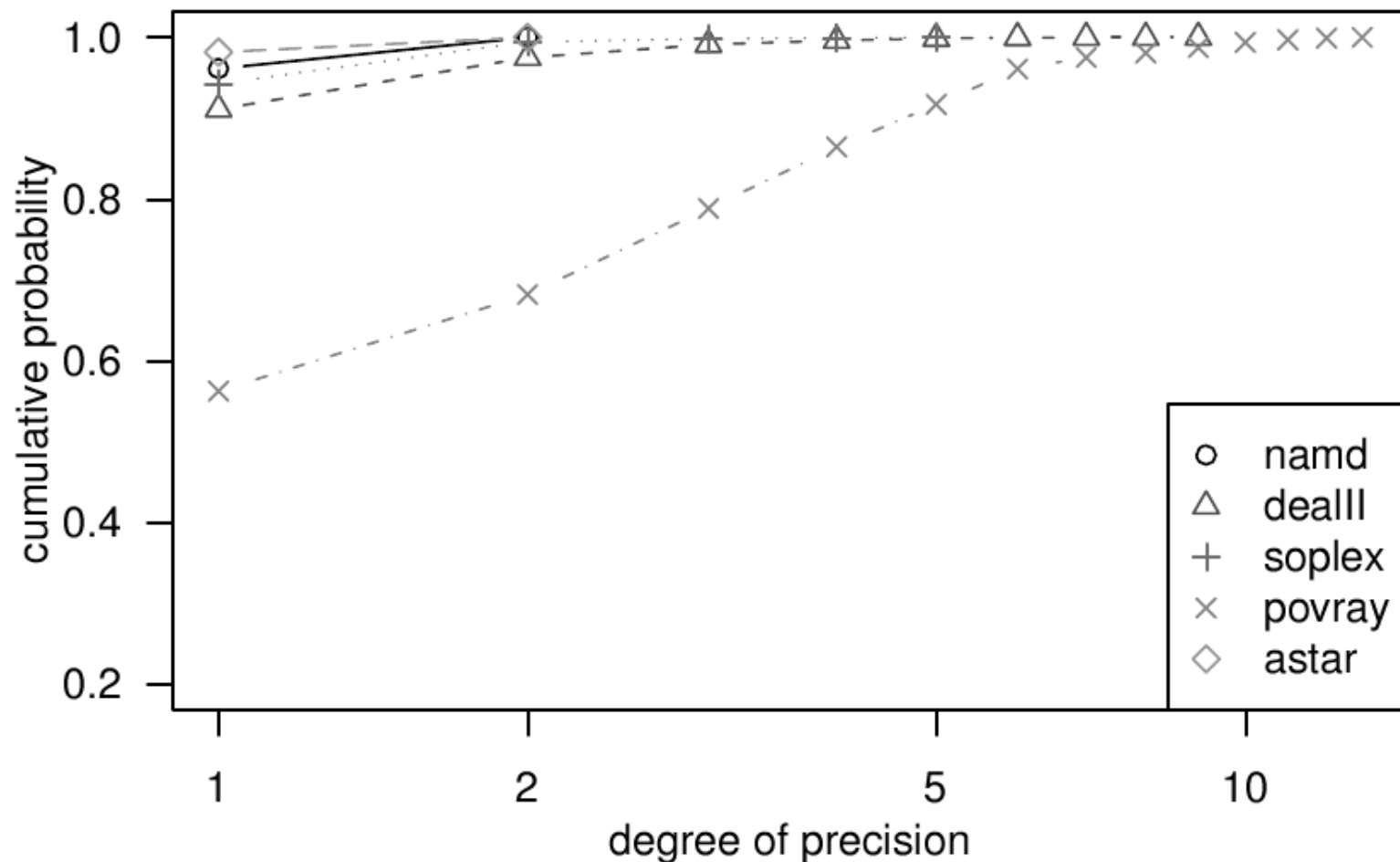
SPEC C



C++: 88% precise

C: 96% precise

Results: Offline Scenario (II)

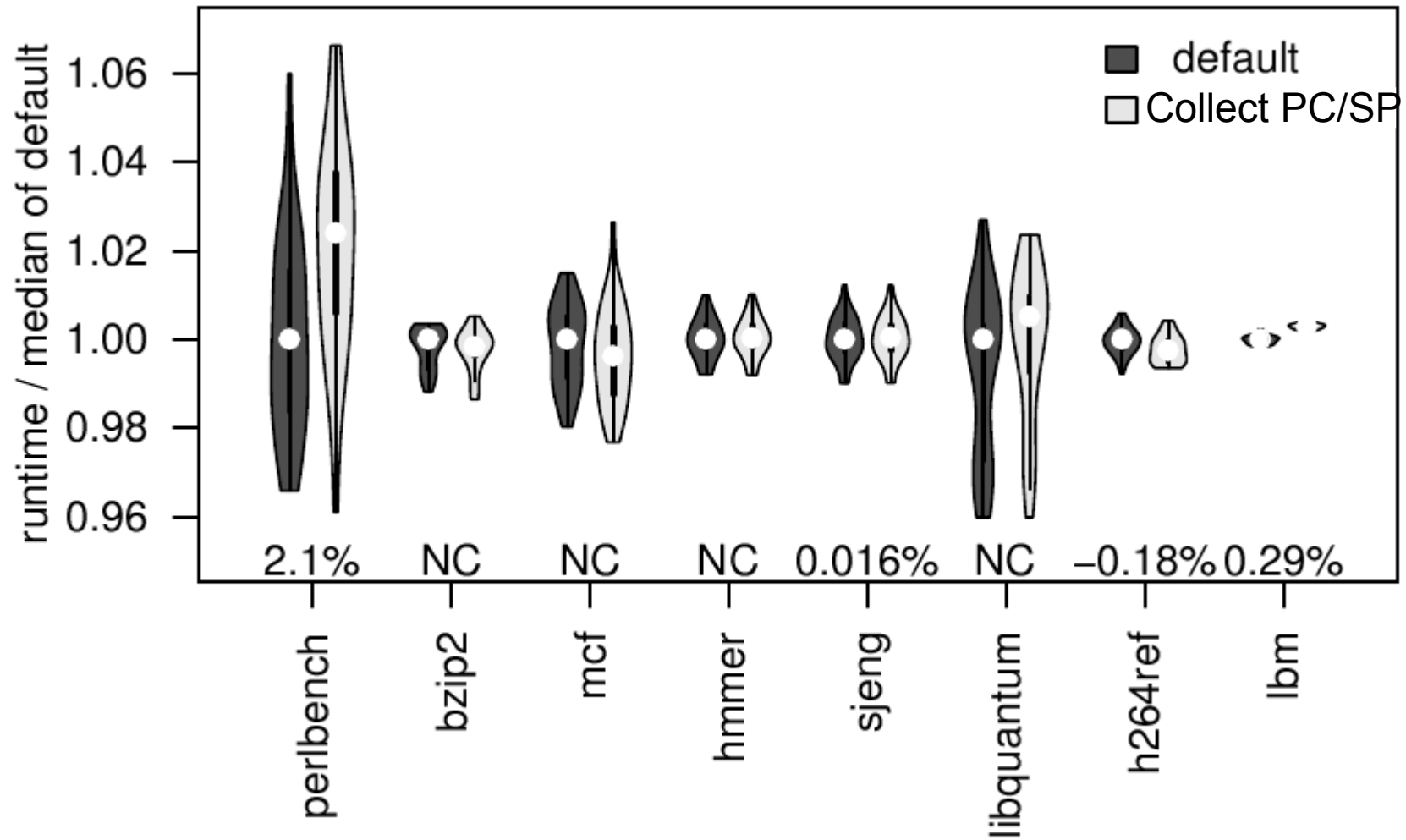


Across SPEC: 5-precise or less = 99%

Results: Online Scenario

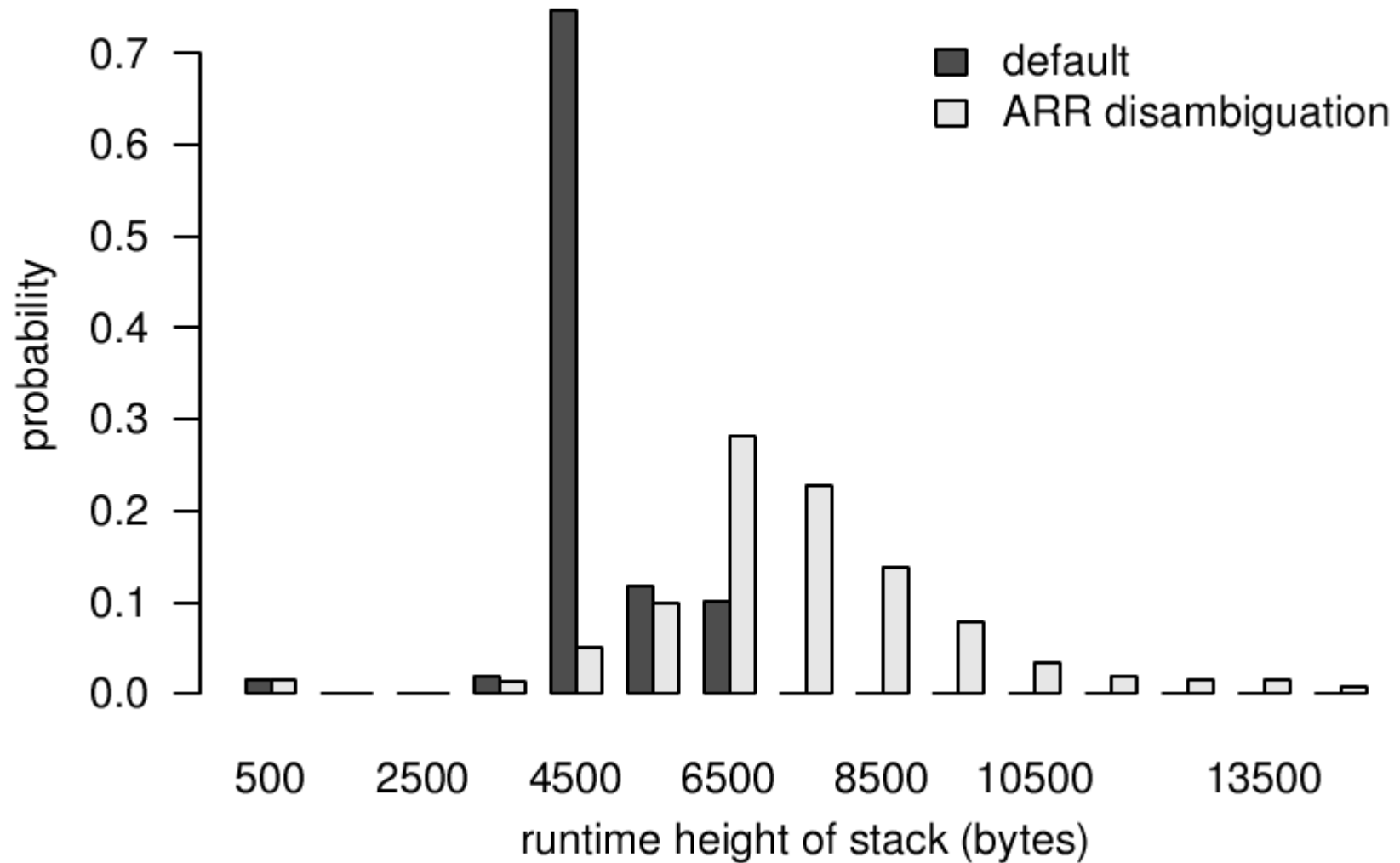
- Unable to find PC/SP in train map
 - 23% for C++
 - 32% for C
- Of those PC/SP pairs in our train map:
 - 80% precise for the C++ programs
 - 91% precise for the C programs
 - 92% PC/SP pairs are 5-precise or less
 - 2% of the time: incorrect

Runtime Overhead



Negligible overhead (geomean = 0.2%) ¹⁷

Space Overhead



Some Space overhead but we can fix

