



# Enabling Compositionality for Multicore Timing Analysis

Jan Reineke @

SAARLAND  
UNIVERSITY



COMPUTER SCIENCE

Joint work with Sebastian Hahn and Michael Jacobs,  
to appear in RTNS 2016.

*Verimag, Grenoble, September 13, 2016*



## Common View

Multicore timing analysis is **hard because of interference** on shared resources.

## But: Timing Analysis for Single Cores

### **WCET Analysis**

= computes bound on execution time of task in **isolation**

### **Response-Time Analysis**

= computes bound on response-time of task accounting for **interference** on shared resource

$$R_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j$$




# Natural Extension to Multicores

## **WCRD Analysis**

= computes bound on resource  
demand of task in **isolation**

Resources: CPU, Bus, Cache, ...



## **Response-Time Analysis**

= computes bound on response-time  
of task accounting for **interference** on  
shared resources



Example:

Shared Resources: Cores + Bus

$C_i$  = worst-case execution time of task  $i$  in isolation

$\sigma_i$  = worst-case number of bus accesses

$B$  = latency of individual bus access

$$R_i = C_i + \left( \sum_{j \in hp(i) \cap \Gamma_i} \left\lceil \frac{R_j}{T_j} \right\rceil \cdot C_j \right) + \left( \sum_{j \in hp(i)} \left\lceil \frac{R_j + J_j}{T_j} \right\rceil \cdot (\sigma_j \cdot B) \right)$$

**worst-case  
interference on core**

**worst-case  
interference on bus**

*(unrealistic simplifying assumption:  
preemptive execution of bus accesses)*



## Challenges

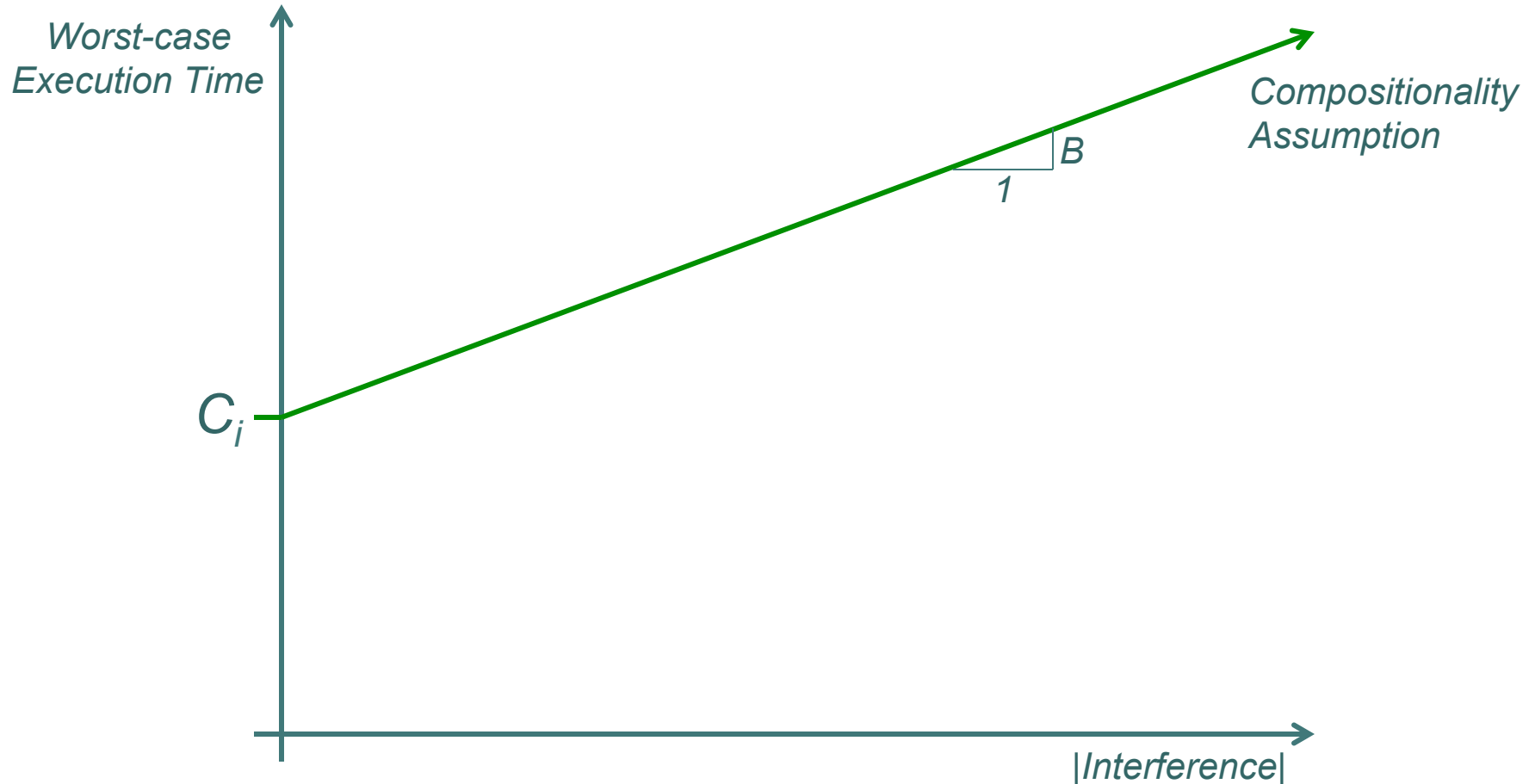
### ***Soundness:***

What is the worst-case cost of being blocked on the bus by one other access?

### ***Precision:***

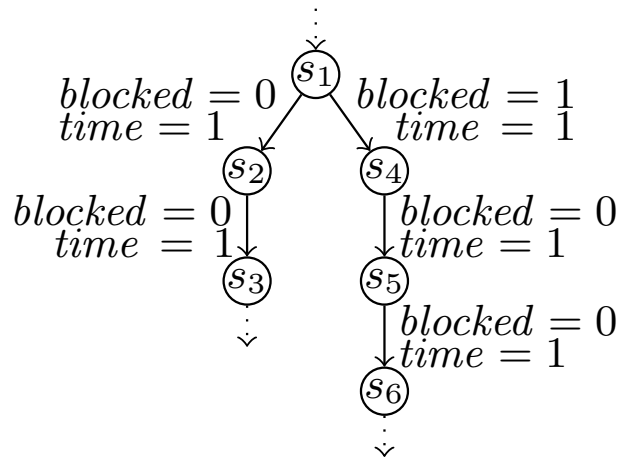
*Isn't execution on different resources overlapped?*

# Interference Response Curves: Compositionality Assumption



# Computation of Actual Interference Response Curves

## Microarchitectural Analysis:



*Model uncertainty about interference by non-determinism.*

## Implicit Path Enumeration:

$$\max \sum_{\text{edge } e} time_e \cdot x_e.$$

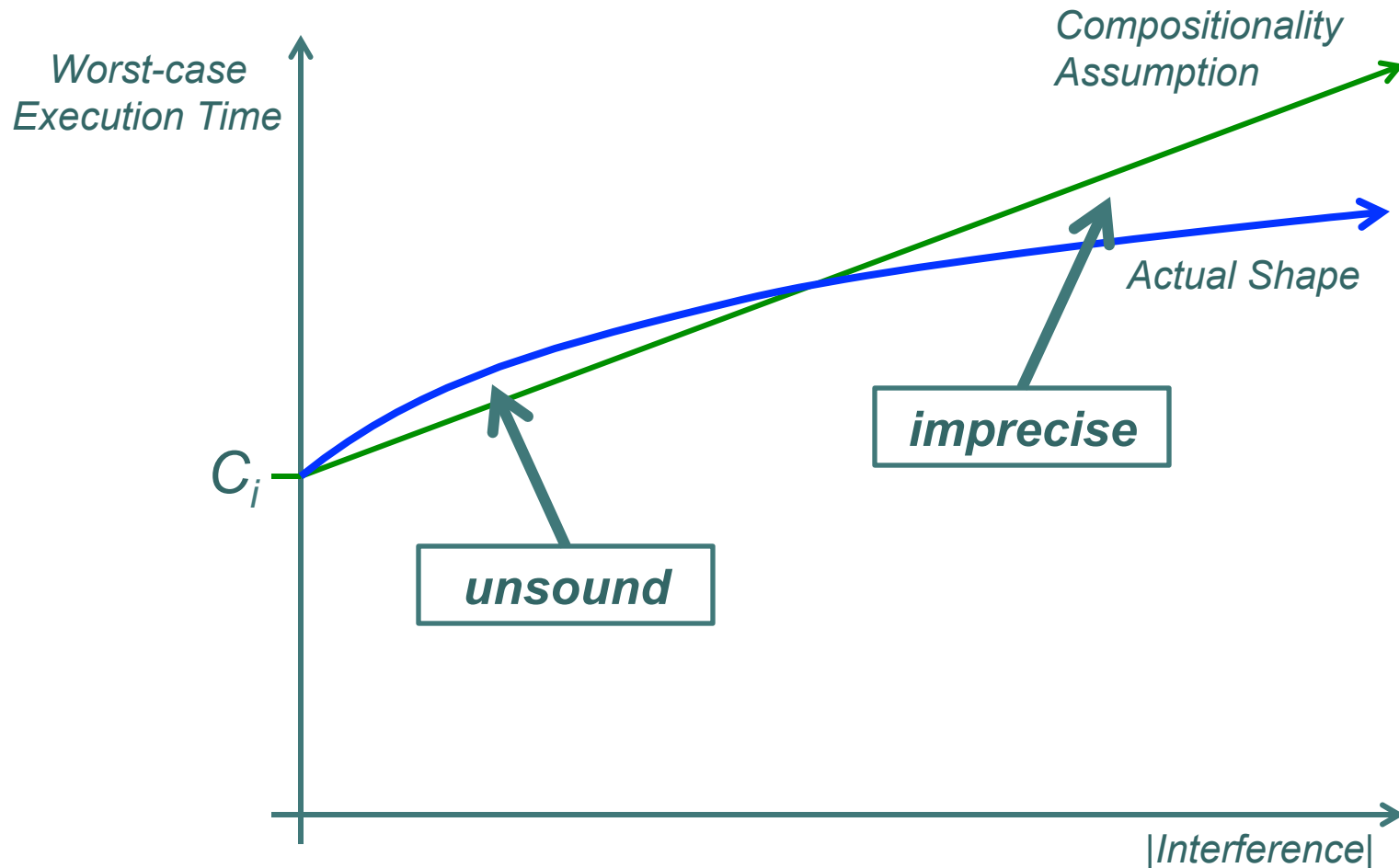
*Flow constraints +  
Loop bounds +  
Feasible paths +*

$$\sum_{\text{edge } e} blocked_e \cdot x_e \leq I$$

***plug in different constants to  
get different points on curve***



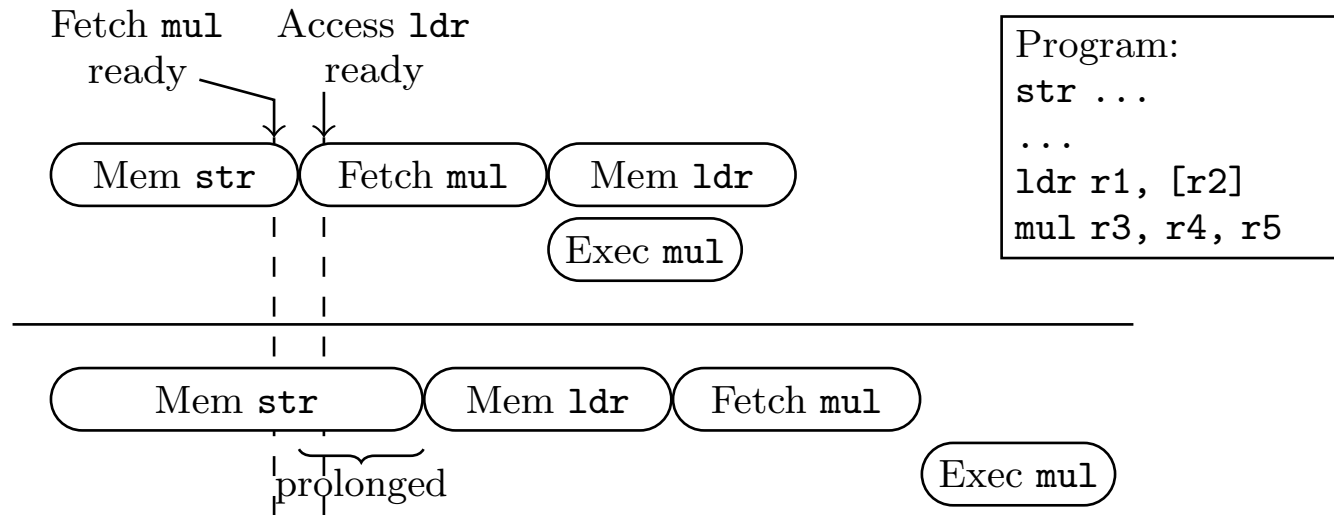
# Interference Response Curves: Actual Shape!



# Why this shape?

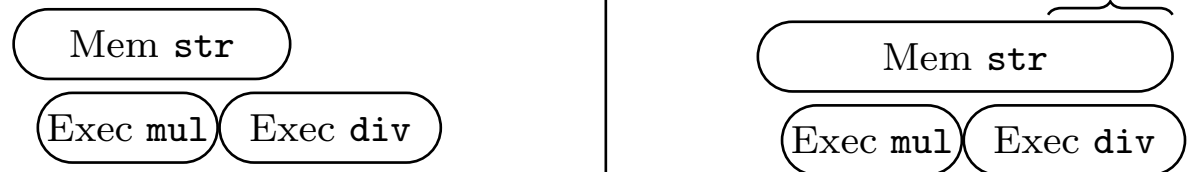
*In some rare cases:*

## *Amplifying Timing Anomaly:*



*But mostly:*

## *Latency Hiding:*



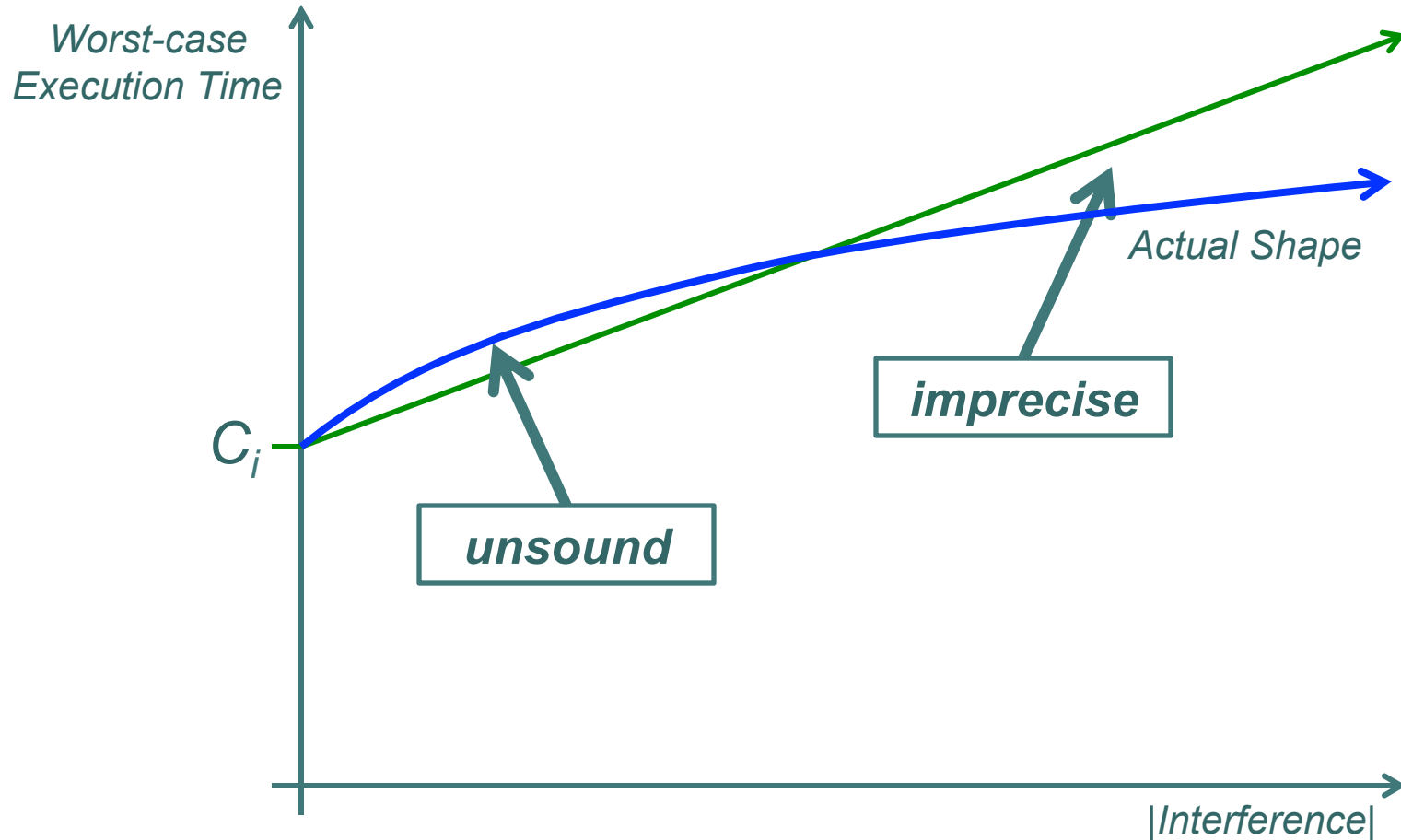


## What to do about it?

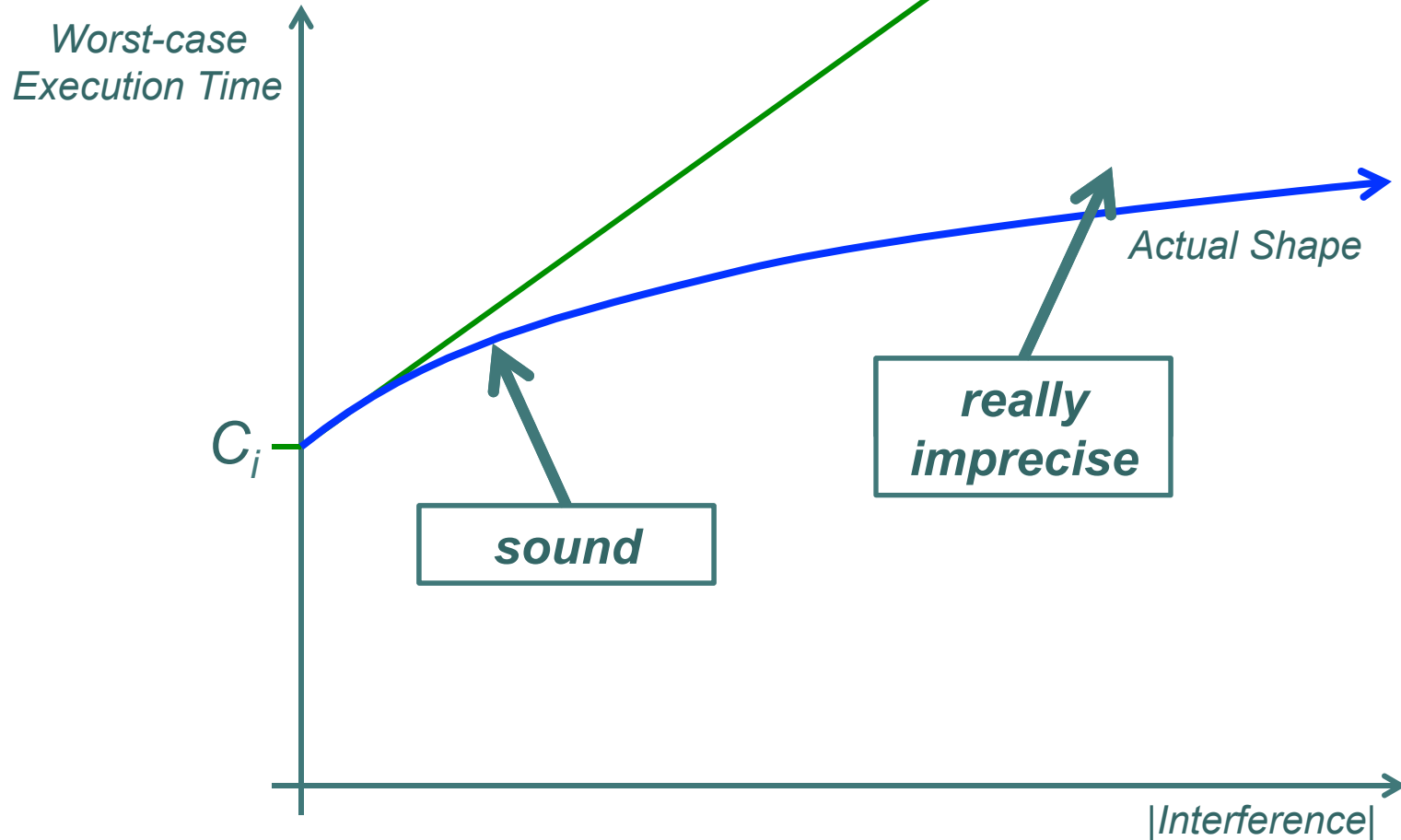
Enabling compositionality:

- 1) (Modify HW)
- 2) More conservative access penalty
- 3) More conservative base bound

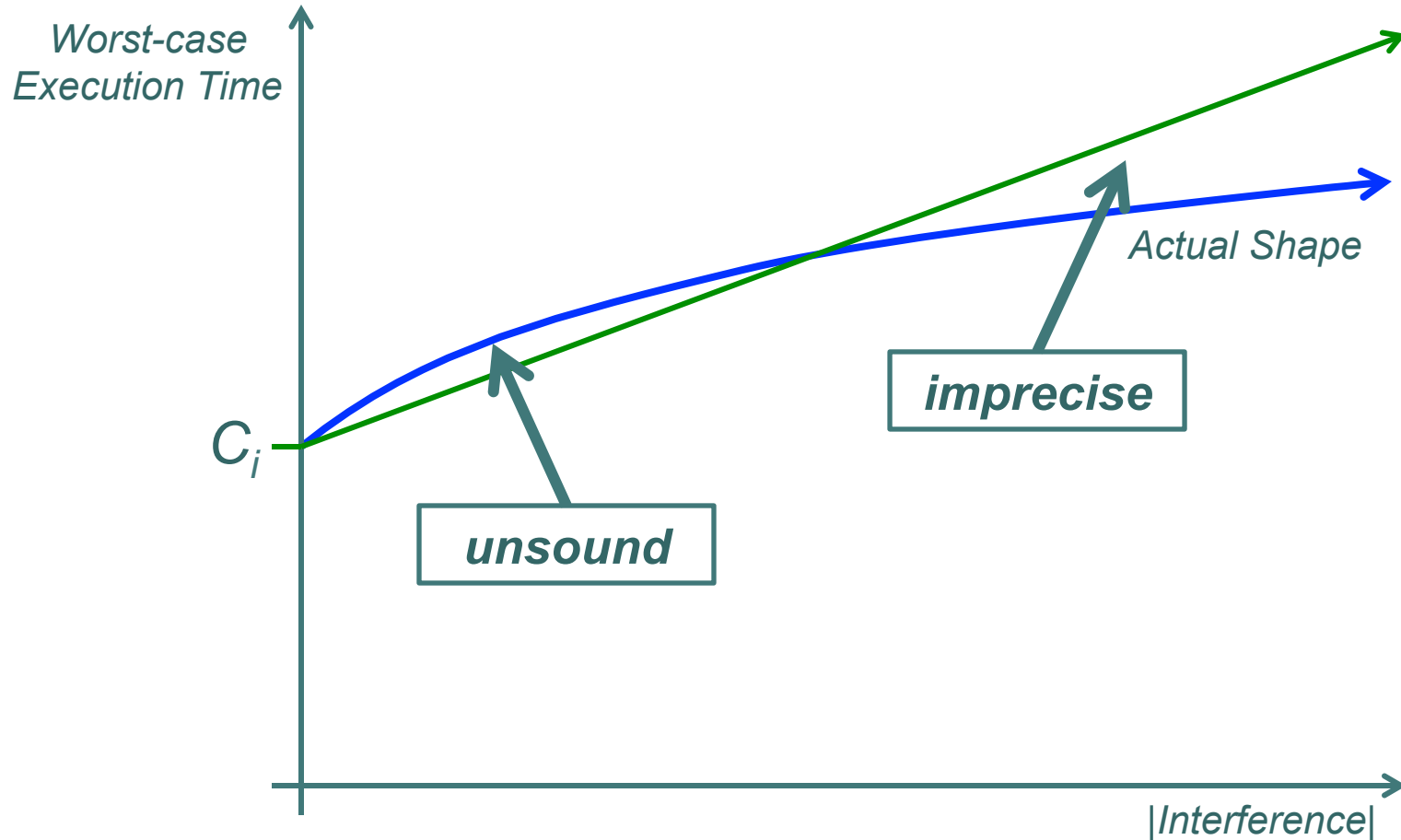
# Enabling Compositionality: Sound Penalty



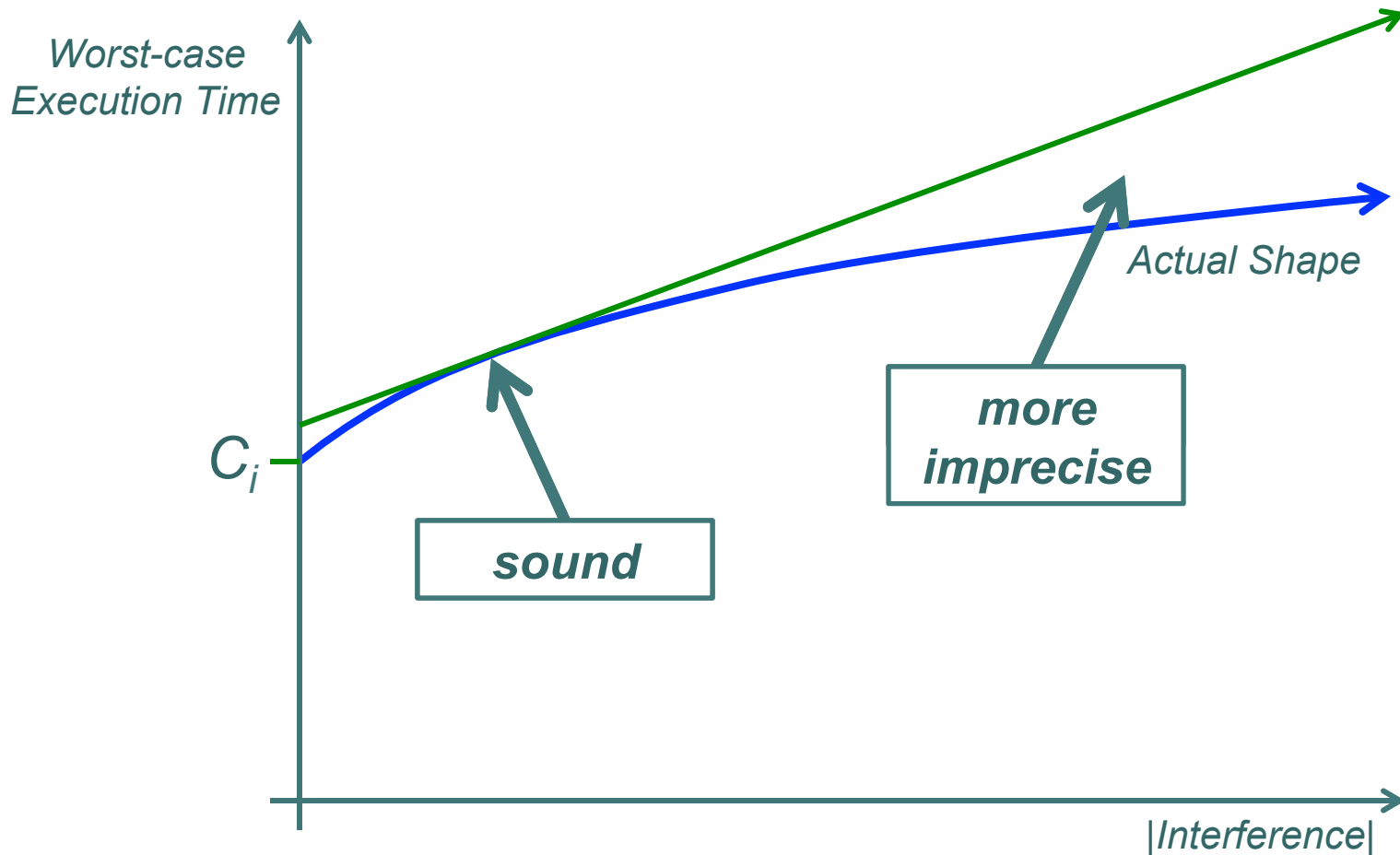
# Enabling Compositionality: Sound Penalty



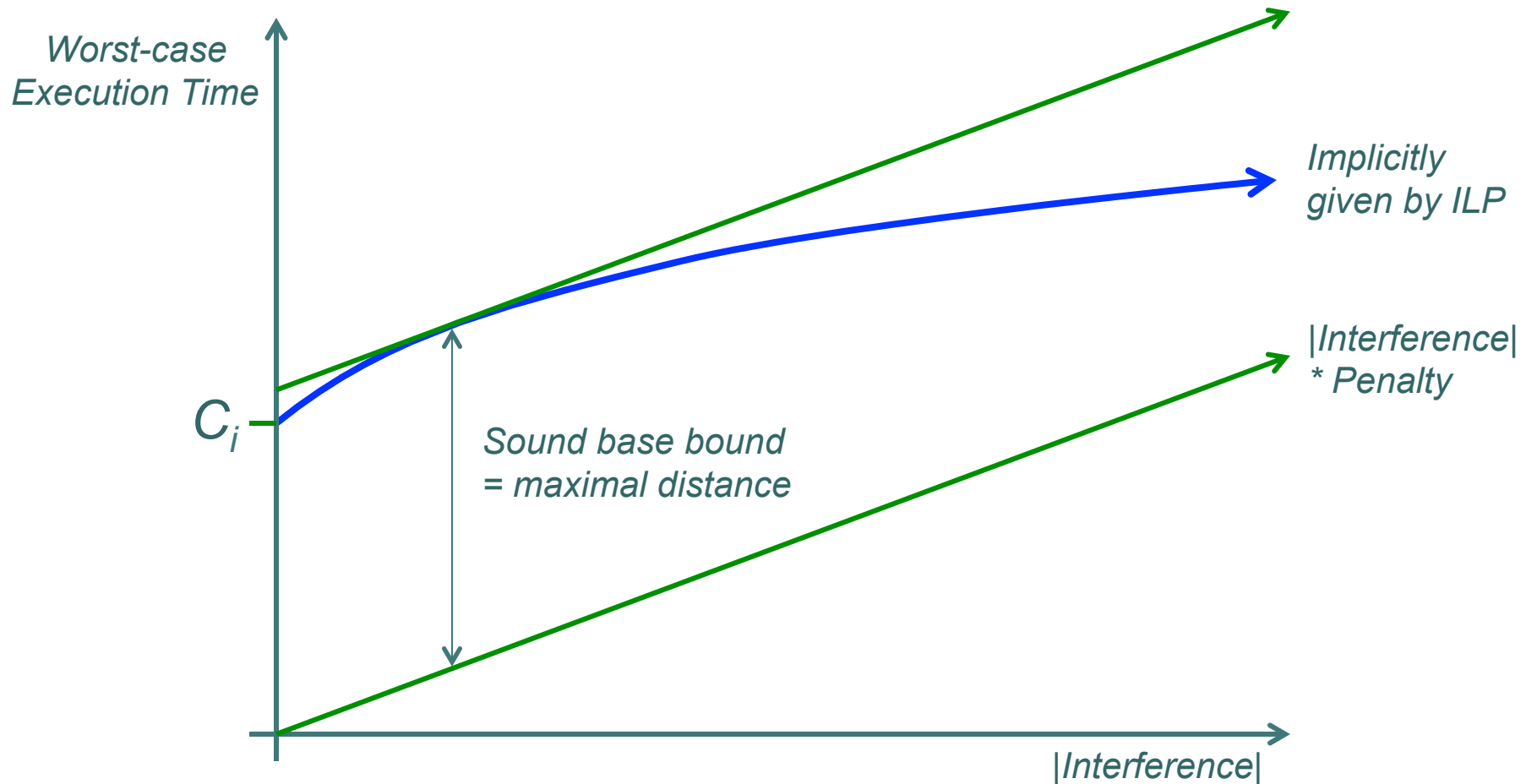
# Enabling Compositionality: Sound Base Bound



# Enabling Compositionality: Sound Base Bound



# How to Compute Sound Base Bound: Intuitively



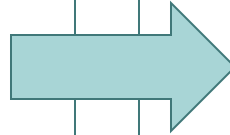


# How to Compute Sound Base Bound: As an ILP

$$\max \sum_{\text{edge } e} time_e \cdot x_e.$$

*Flow constraints +  
Loop bounds +  
Feasible paths +*

$$\sum_{\text{edge } e} blocked_e \cdot x_e \leq I$$



$$\max \left( \sum_{\text{edge } e} time_e \cdot x_e \right) - bp \cdot ba.$$

*Flow constraints +  
Loop bounds +  
Feasible paths +*

$$\sum_{\text{edge } e} blocked_e \cdot x_e \leq ba.$$

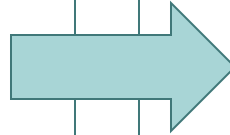
***variable rather than constant***

# How to Compute Sound Base Bound: As an ILP

$$\max \sum_{\text{edge } e} time_e \cdot x_e.$$

*Flow constraints +  
Loop bounds +  
Feasible paths +*

$$\sum_{\text{edge } e} blocked_e \cdot x_e \leq I$$



$$\max \left( \sum_{\text{edge } e} time_e \cdot x_e \right) - bp \cdot ba.$$

*Flow constraints +  
Loop bounds +  
Feasible paths +*

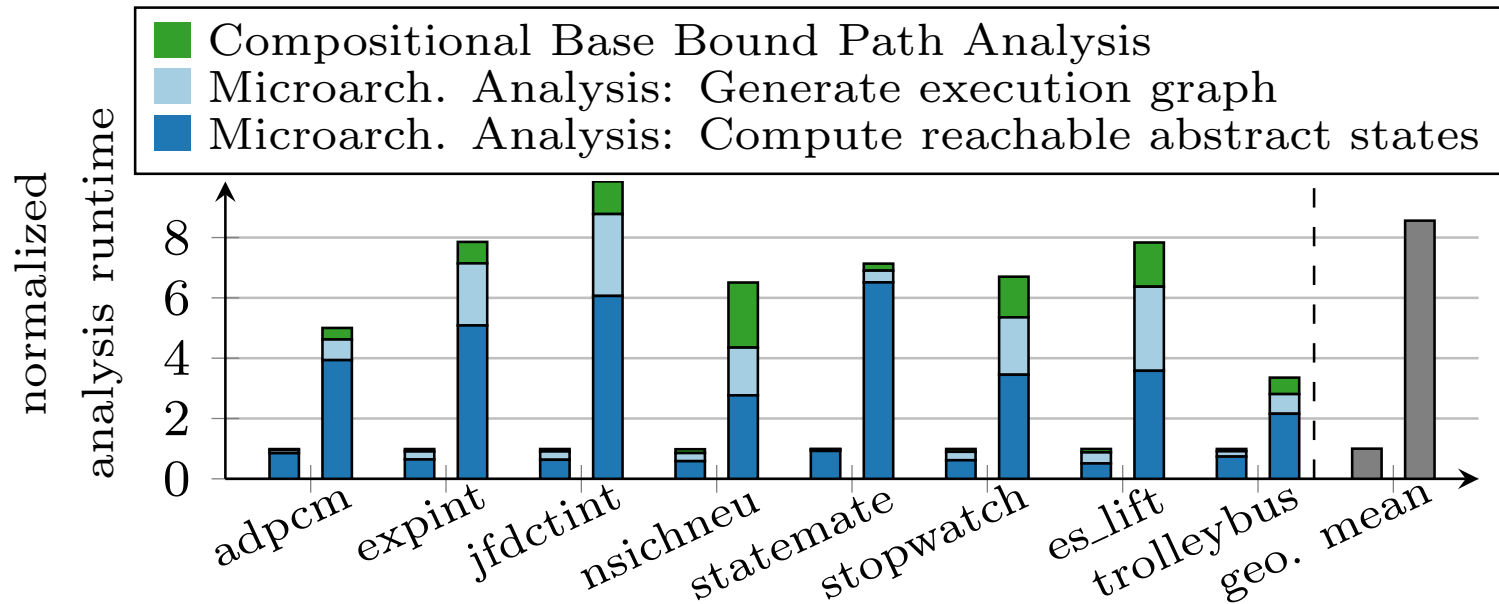
$$\sum_{\text{edge } e} blocked_e \cdot x_e \leq ba.$$

***variable rather than constant***

*Can be generalized to multiple dimensions: e.g. bus interference + CRPD + refreshes*

# Some Experimental Results:

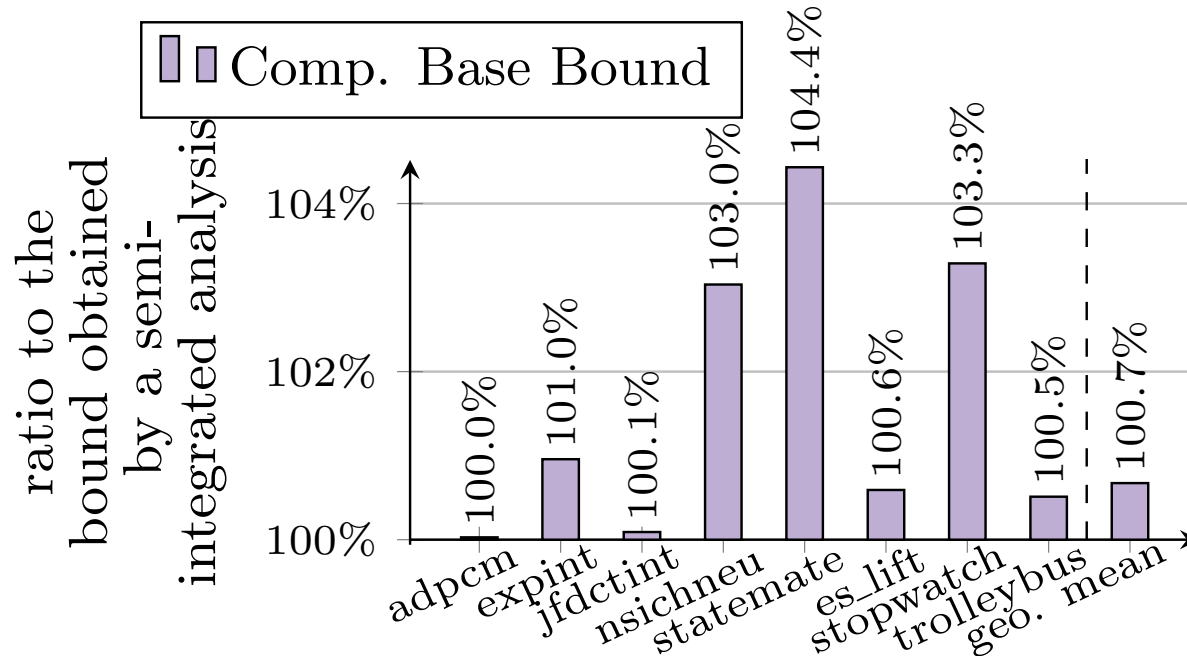
## Analysis Cost: Sound Base Bound



*For quad-core out-of-order processor with store buffer.*

# Some Experimental Results:

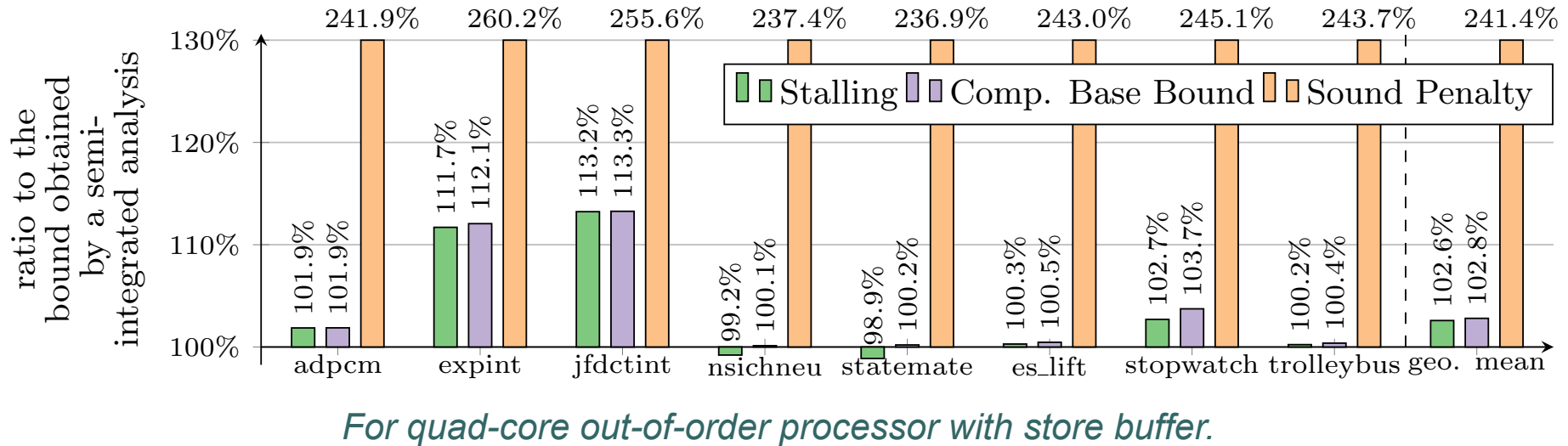
## Analysis Precision: No Interference



*For quad-core out-of-order processor with store buffer.*

# Some Experimental Results:

## Analysis Precision: 100% Interference





## Conclusions

**Sound base bound** enables compositional response-time analysis:

- ~ 8x analysis slowdown (depends heavily on benchmark and processor configuration)
- relatively small imprecision