

Application Slowdown Model

Quantifying and Controlling Impact of
Interference at Shared Caches and Main Memory

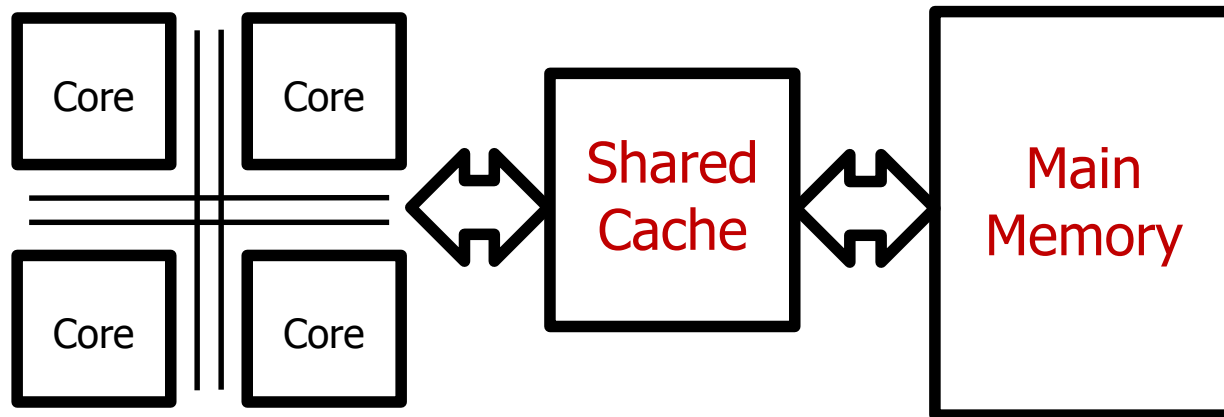
Lavanya Subramanian, Vivek Seshadri,
Arnab Ghosh, Samira Khan, Onur Mutlu

SAFARI

Carnegie Mellon

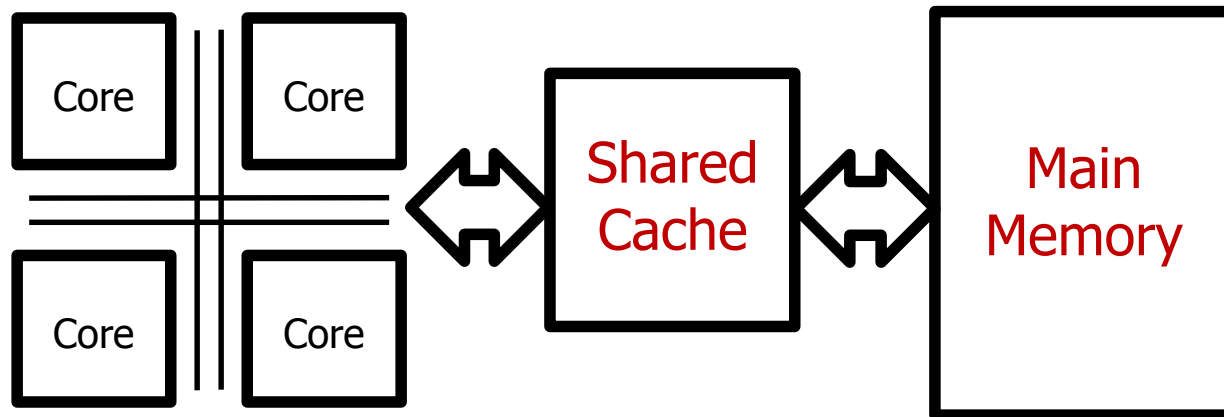


Problem: Interference at Shared Resources



- 1. High application slowdowns*
- 2. Unpredictable application slowdowns*

Problem: Interference at Shared Resources



Our Goal: Achieve high and predictable performance

Our Approach

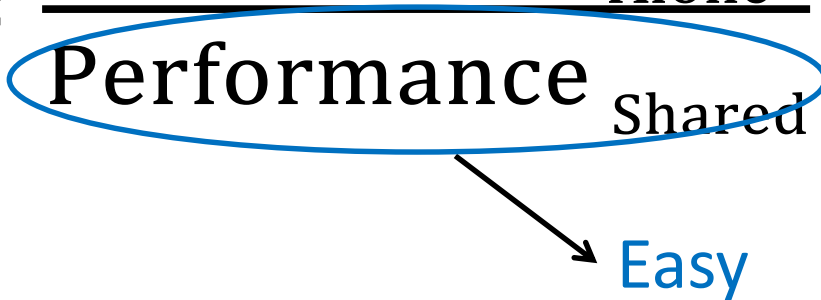
1. Build a model to accurately estimate *slowdowns*
2. Use slowdown estimates to build slowdown-aware resource management mechanisms

Challenge in Estimating Slowdown

$$\text{Slowdown} = \frac{\text{Performance}_{\text{Alone}}}{\text{Performance}_{\text{Shared}}}$$

Challenge in Estimating Slowdown

$$\text{Slowdown} = \frac{\text{Performance}_{\text{Alone}}}{\text{Performance}_{\text{Shared}}}$$



Challenge in Estimating Slowdown

$$\text{Slowdown} = \frac{\text{Performance}_{\text{Alone}}}{\text{Performance}_{\text{Shared}}}$$

Challenge

Easy

The diagram illustrates the formula for Slowdown, which is the ratio of Performance Alone to Performance Shared. The numerator, 'Performance Alone', is circled in red and has an arrow pointing to the word 'Challenge' in red text. The denominator, 'Performance Shared', is circled in blue and has an arrow pointing to the word 'Easy' in blue text.

Our Model

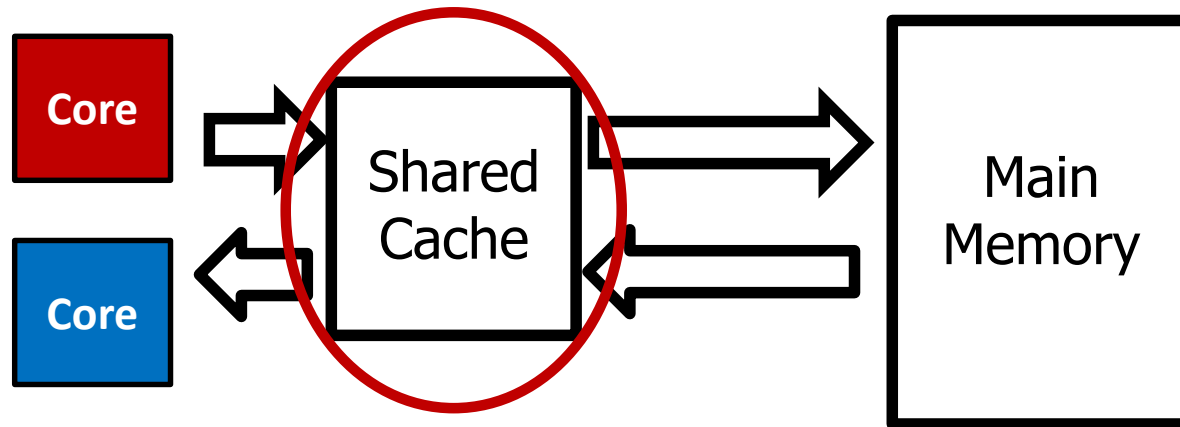
Our model overcomes this challenge

Our estimation error: 10%

Best previous model's error: 30%

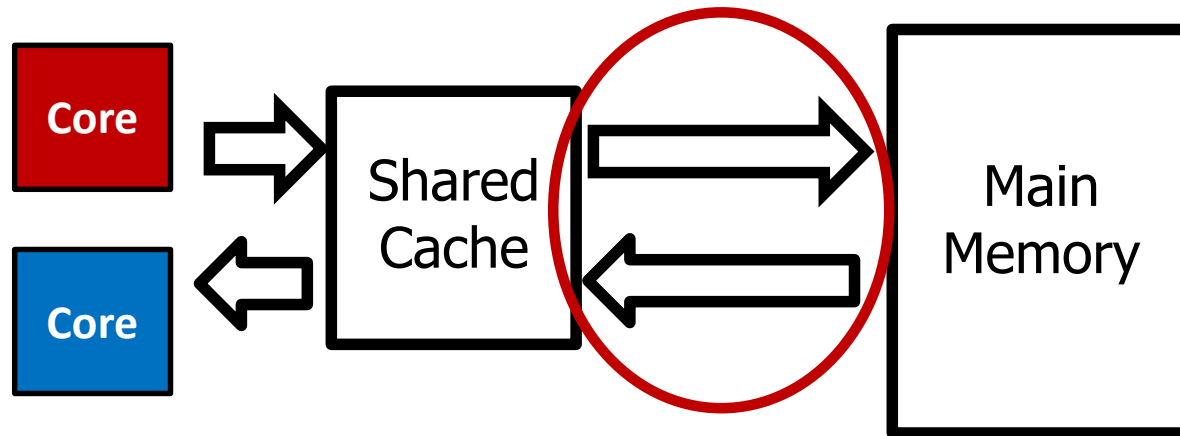
Leveraging Our Slowdown Estimates

Leveraging Our Slowdown Estimates



*Slowdown-aware
cache capacity partitioning*

Leveraging Our Slowdown Estimates



*Slowdown-aware
memory bandwidth partitioning*

Talk at 2:40pm in Tapa Ballroom 2

Application Slowdown Model

**Lavanya Subramanian, Vivek Seshadri,
Arnab Ghosh, Samira Khan, Onur Mutlu**

SAFARI

Carnegie Mellon

