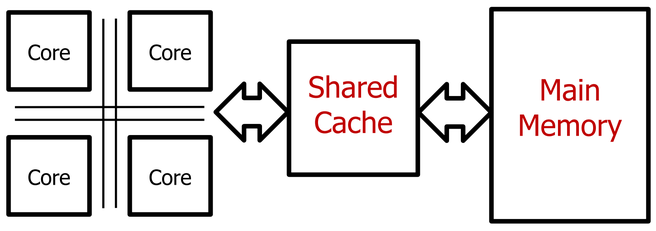
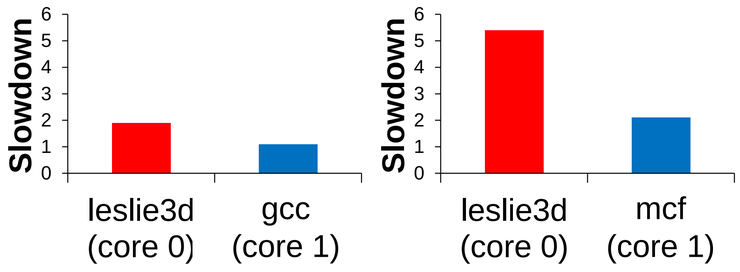
Application Slowdown Model:
Quantifying and Controlling Impact of Interference at Shared Caches and Main Memory

Lavanya Subramanian, Vivek Seshadri, Arnab Ghosh, Samira Khan, Onur Mutlu
Carnegie Mellon University, Intel Labs, University of Virginia

Problem: Shared Resource Interference



Impact of Shared Resource Interference



1. High application slowdowns
2. Unpredictable application slowdowns

Our Goal

Provide high and predictable performance in the presence of shared resource interference

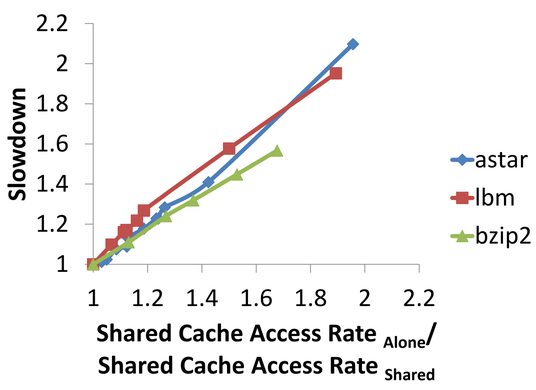
Our Approach

1. Build a model to estimate slowdowns
2. Leverage our model for slowdown-aware resource management

Application Slowdown Model (ASM)

Observation: Proxy for Performance

For a memory bound application,
Performance α Cache access rate

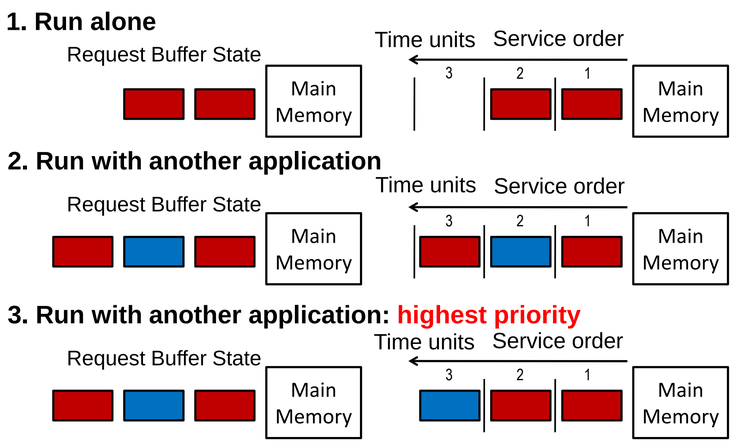


$$\text{Slowdown} = \frac{\text{Performance}_{\text{Alone}}}{\text{Performance}_{\text{Shared}}}$$

$$\text{Slowdown} = \frac{\text{Cache Access Rate}_{\text{Alone}}\ (\text{CAR}_{\text{Alone}})}{\text{Cache Access Rate}_{\text{Shared}}\ (\text{CAR}_{\text{Shared}})}$$

Challenge: Estimating Cache Access Rate $_{\text{Alone}}$

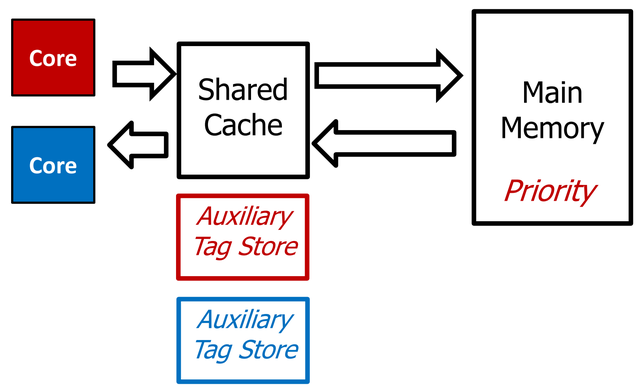
Minimize memory bandwidth contention:
Using priority (Subramanian et al., HPCA 2013)



Highest priority → Little interference
(almost as if application were run alone)

*Enables estimation of **miss service time***

Quantify shared cache capacity contention:
Using auxiliary tag stores (Pomerene et al., 1989)



Auxiliary tag store counts *#contention misses*

Cache Contention Cycles = #Contention Misses x Average Miss Service Time

From auxiliary tag store when given high priority

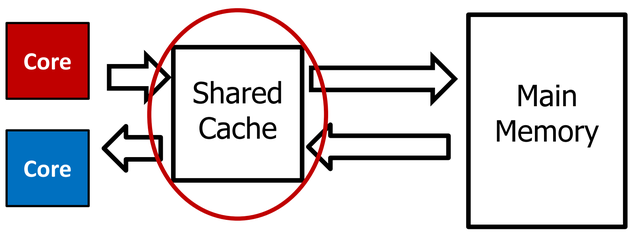
Measured when given high priority

Remove contention cycles when estimating CAR_{Alone}

Average error of ASM: 10%; Average error of previous models: 30%

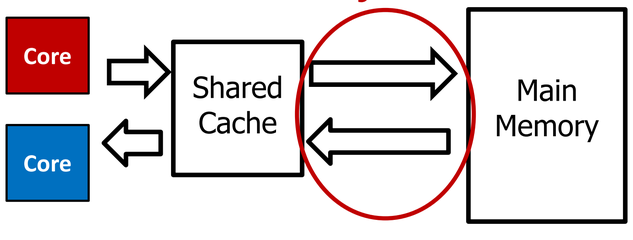
Leveraging the Application Slowdown Model

Slowdown-aware cache capacity partitioning



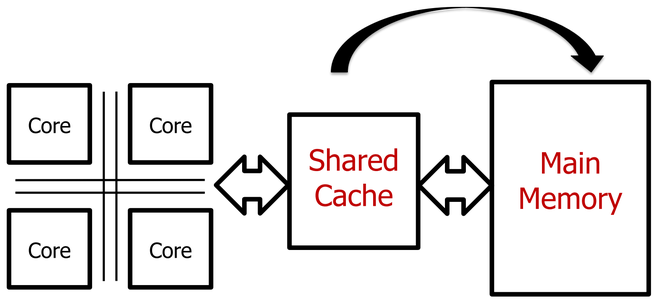
Previous work: Reduce miss counts;
Our proposal: Reduce slowdowns

Slowdown-aware memory bandwidth partitioning

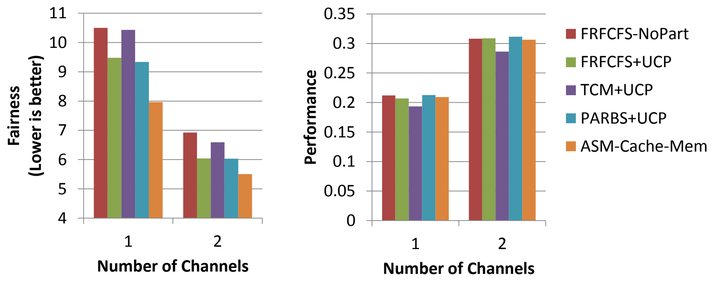


Allocation memory bandwidth proportional to slowdowns

Coordinated Resource Allocation Schemes



16-core system 100 workloads



Significant fairness benefits across different channel counts

Providing Slowdown Guarantees

- *Cache allocation with the goal of meeting a slowdown bound*
- Allocate just enough cache space to critical application
- Allocate remaining cache space to other applications