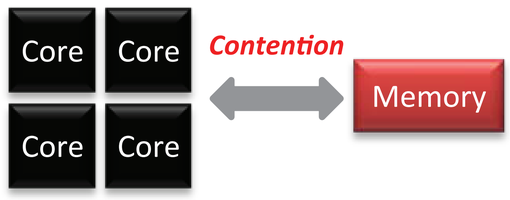
THREAD CLUSTER MEMORY SCHEDULING

Yoongu Kim, Michael Papamichael, Onur Mutlu, Mor Harchol-Balter (CMU)

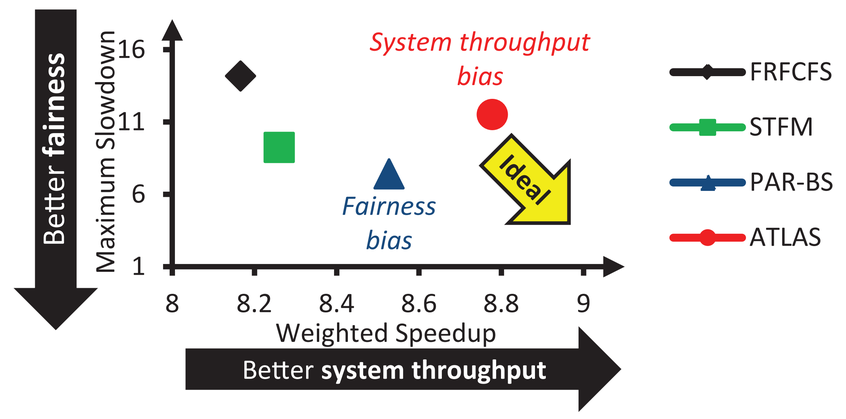
MOTIVATION

- **Memory is a key shared resource in CMPs**
- **Contention for memory access leads to:**
  - **Degradation in single-thread performance**
  - **Starvation**



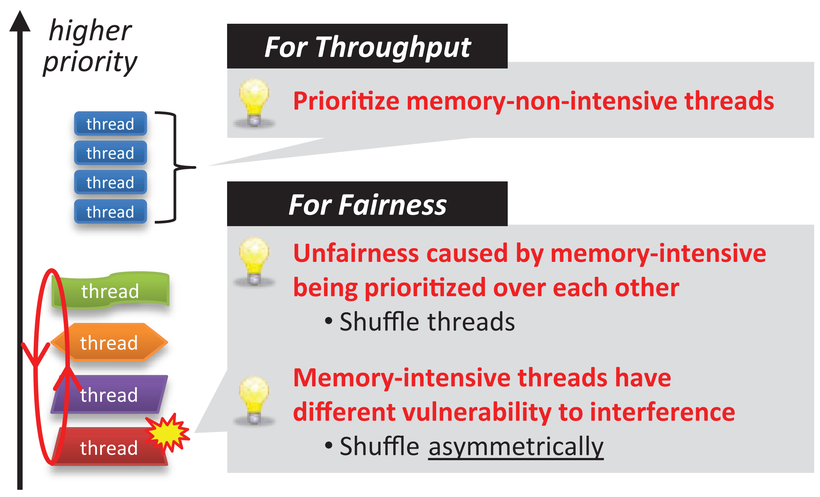
How to achieve both system throughput and fairness?

SHORTCOMINGS OF PREVIOUS WORK



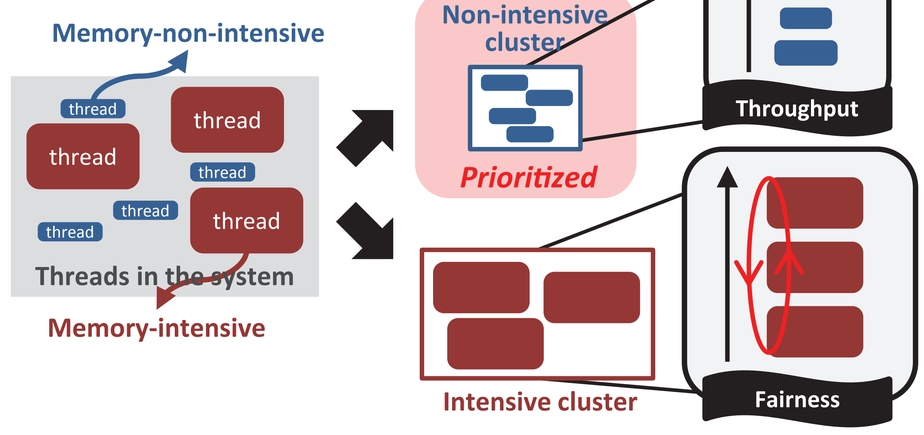
Previous approaches are biased: can't achieve both

INSIGHT: BEST OF BOTH WORLDS



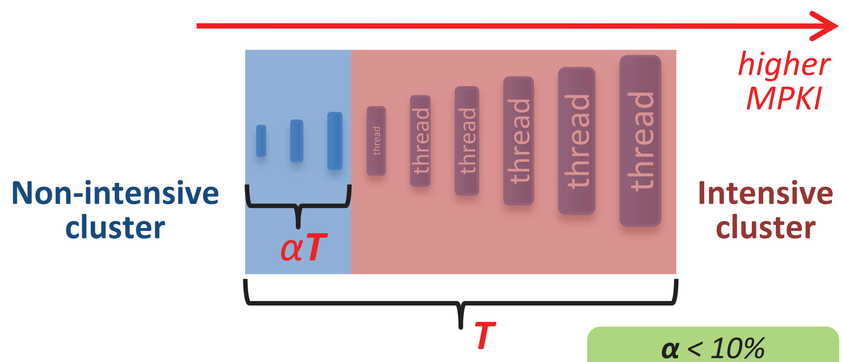
THREAD CLUSTER MEMORY SCHEDULING

1. **Group threads into two clusters**
2. **Prioritize non-intensive cluster**
3. **Different policies for each cluster**



HOW TO CLUSTER THREADS?

Step1 Sort threads by **MPKI** (misses per kilo instruction)



T = Total ***memory bandwidth usage***

α < 10%
ClusterThreshold

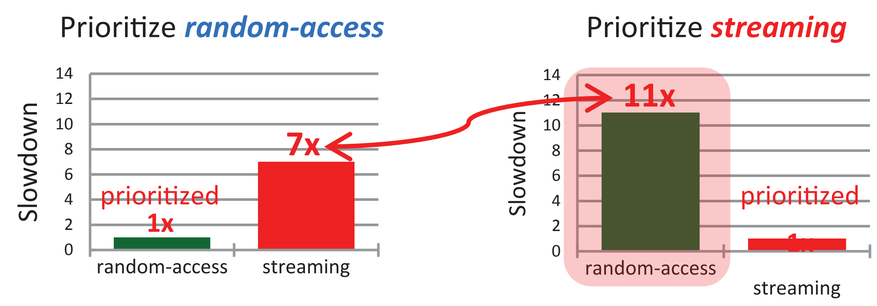
Step2 Memory bandwidth usage ***αT*** divides clusters

PRIORITIZATION WITHIN INTENSIVE CLUSTER - I

Case Study: Two intensive threads contending

1. *random-access*
2. *streaming*

Which is slowed down more easily?



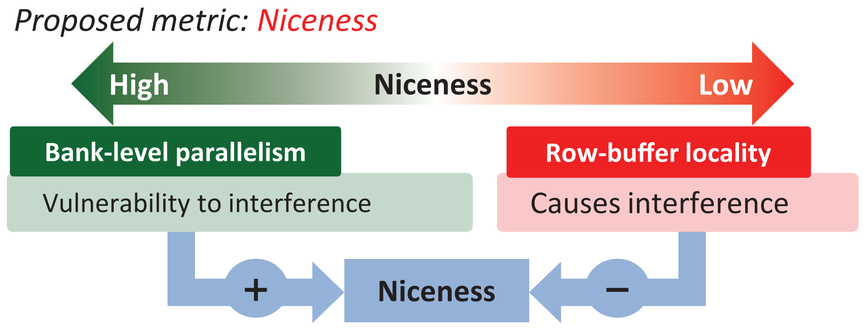
random-access thread is more easily slowed down

- Vulnerable to interference
- Causes less interference to other threads

PRIORITIZATION WITHIN INTENSIVE CLUSTER - II

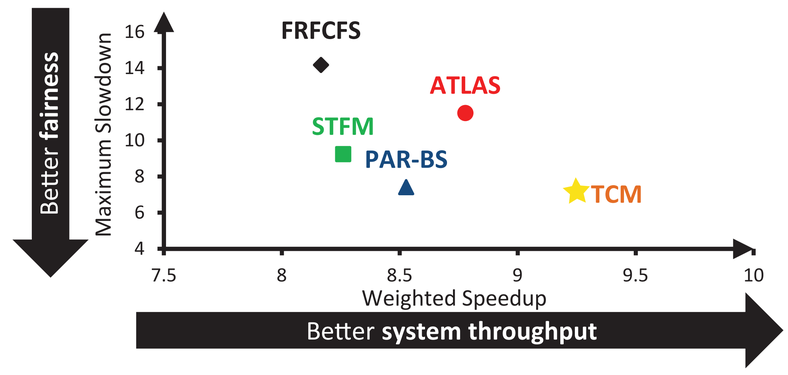
How to quantify difference between threads?

Proposed metric: Niceness



Shuffle priorities in a niceness-aware, asymmetric manner

RESULTS



Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley