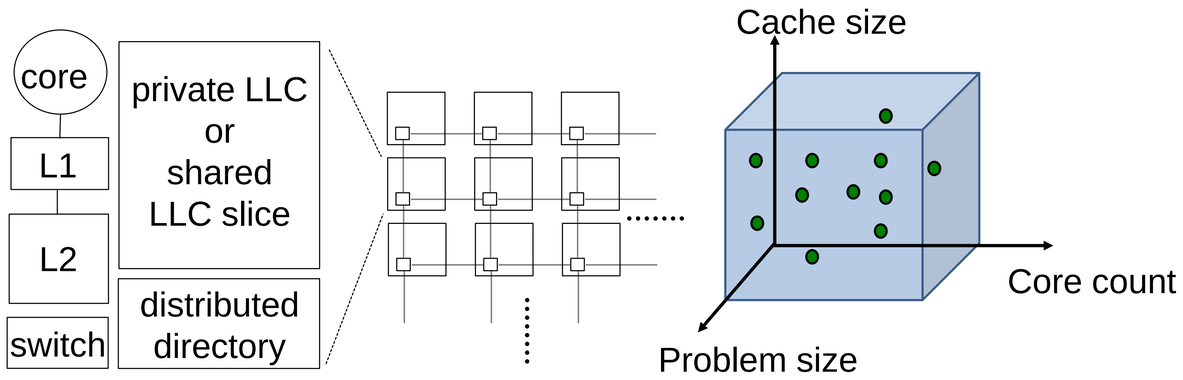
Identifying Optimal Multicore Cache Hierarchies for Loop-based Parallel Programs via Reuse Distance Analysis

Meng-Ju Wu and Donald Yeung
University of Maryland, College Park

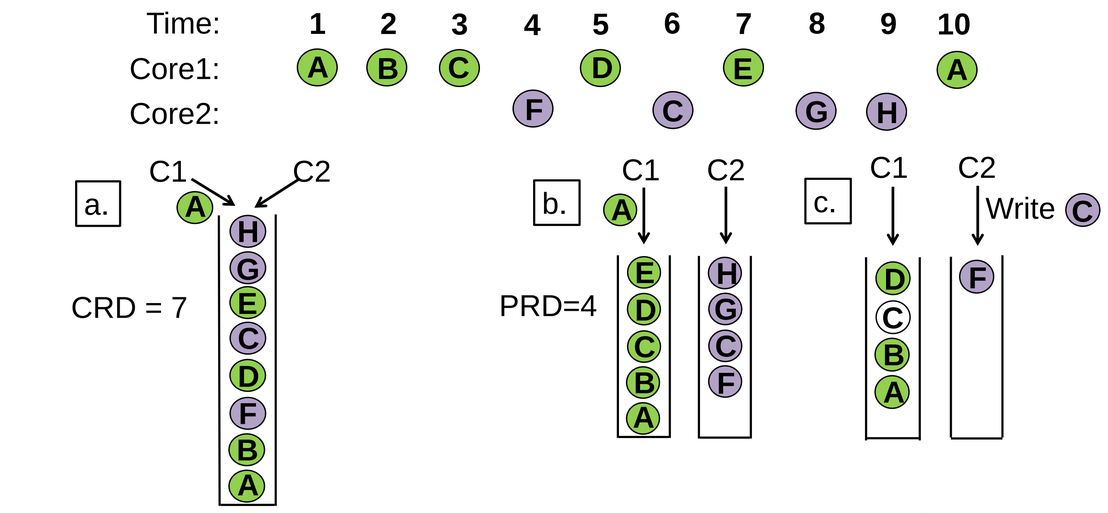
Problem

- Trend → Multicore
- Understanding memory performance is critical but difficult
- Problems: large design space and very slow simulation
  - **3,168** simulations → **3** months on **74** core cluster



Solution

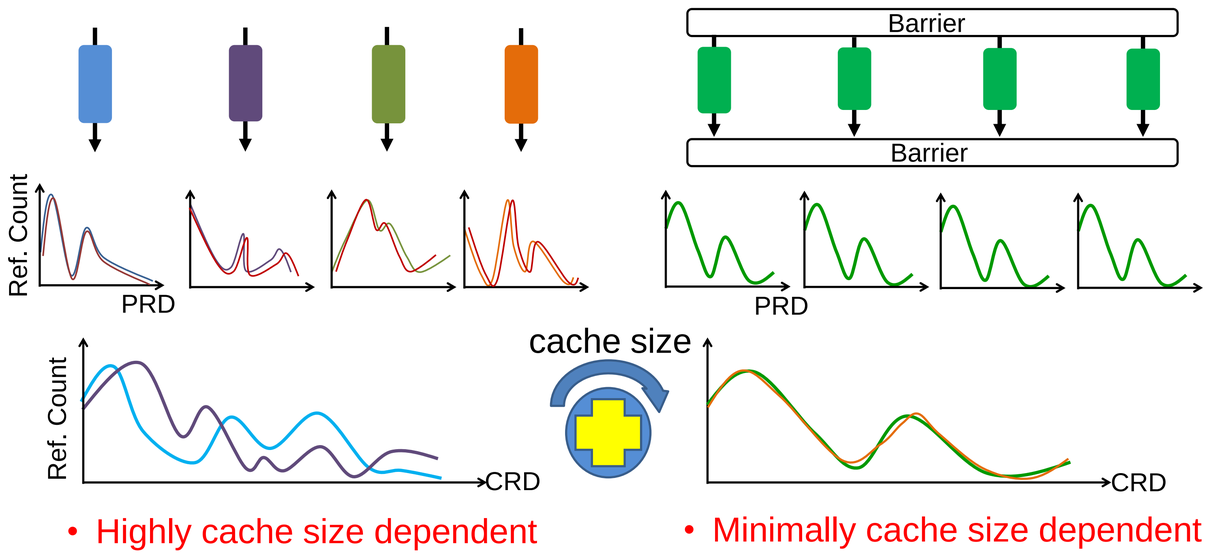
- Multicore Reuse Distance Analysis
- Concurrent Reuse Distance (CRD): shared cache[Ding09,Schuff09,Jiang10]
- Private-stack Reuse Distance (PRD): private cache[Schuff09]



(a) interleave and overlap for CRD (b) replication and (c) invalidation for PRD

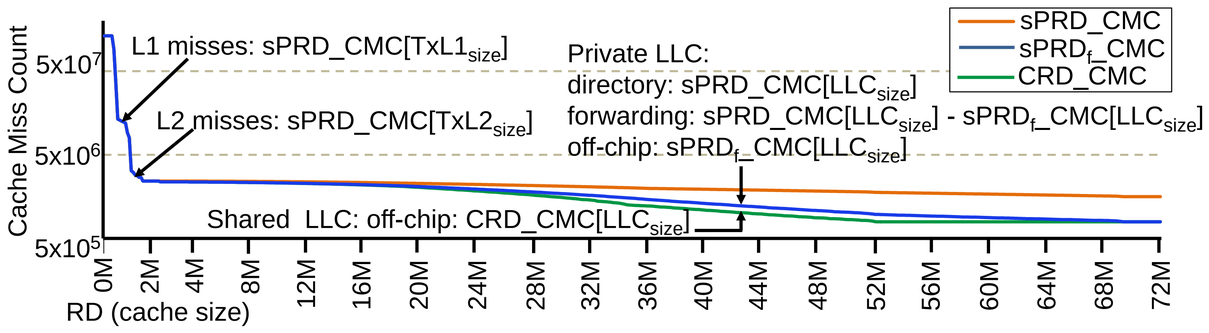
Cache Size Dependency

- Asymmetric threads
- Symmetric threads



- Highly cache size dependent
- Minimally cache size dependent

Cache Performance Models



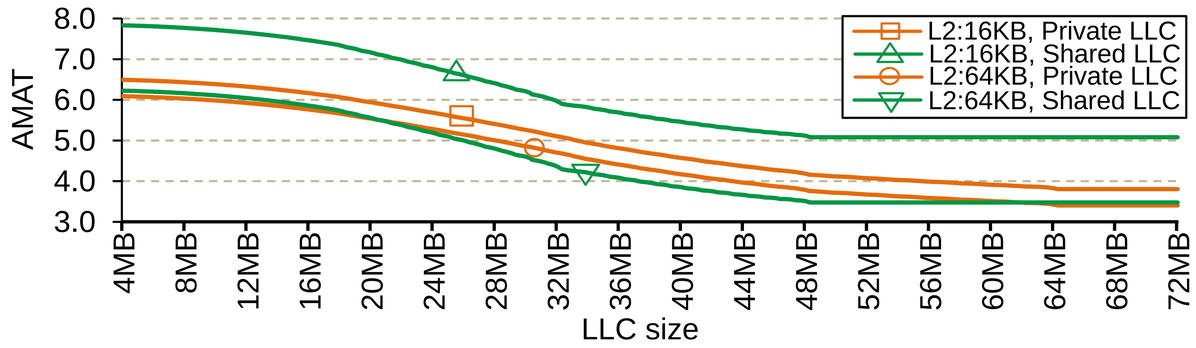
$$AMAT_P = (L1_{lat} \times sPRD\_CMC[0] + L2_{lat} \times sPRD\_CMC[T \times L1_{size}] + LLC_{lat} \times sPRD\_CMC[T \times L2_{size}] + (DIR_{lat} + HOP_{lat}) \times sPRD\_CMC[LLC_{size}] + (LLC_{lat} + 2 \times HOP_{lat}) \times (sPRD\_CMC[LLC_{size}] - sPRD_f\_CMC[LLC_{size}]) + (DRAM_{lat} + 2 \times HOP_{lat}) \times sPRD_f\_CMC[LLC_{size}]) / sPRD\_CMC[0]$$

$$AMAT_S = (L1_{lat} \times sPRD\_CMC[0] + L2_{lat} \times sPRD\_CMC[T \times L1_{size}] + (LLC_{lat} + 2 \times HOP_{lat}) \times sPRD\_CMC[T \times L2_{size}] + (DRAM_{lat} + 2 \times HOP_{lat}) \times CRD\_CMC[LLC_{size}]) / sPRD\_CMC[0]$$

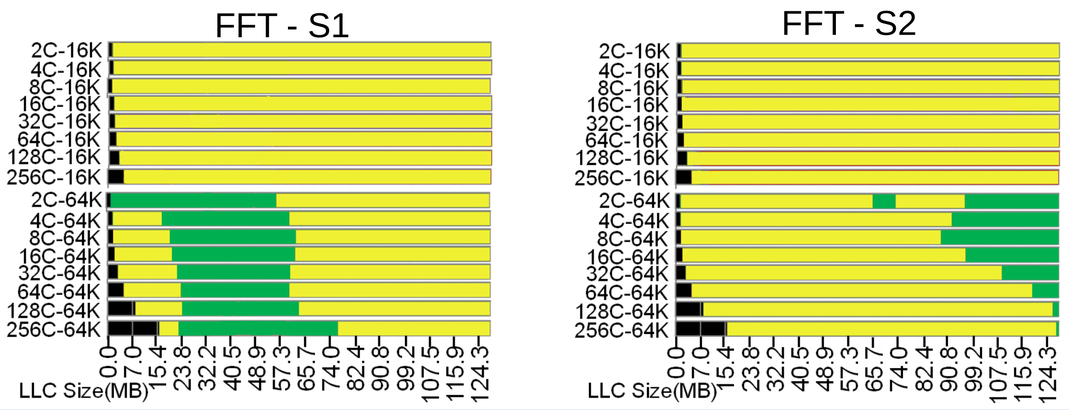
Private vs. Shared LLC

- $AMAT_P > AMAT_S$

  Private LLC's off-chip memory stall - Shared LLC's off-chip memory stall > Shared LLC's on-chip memory stall - Private LLC's on-chip memory stall

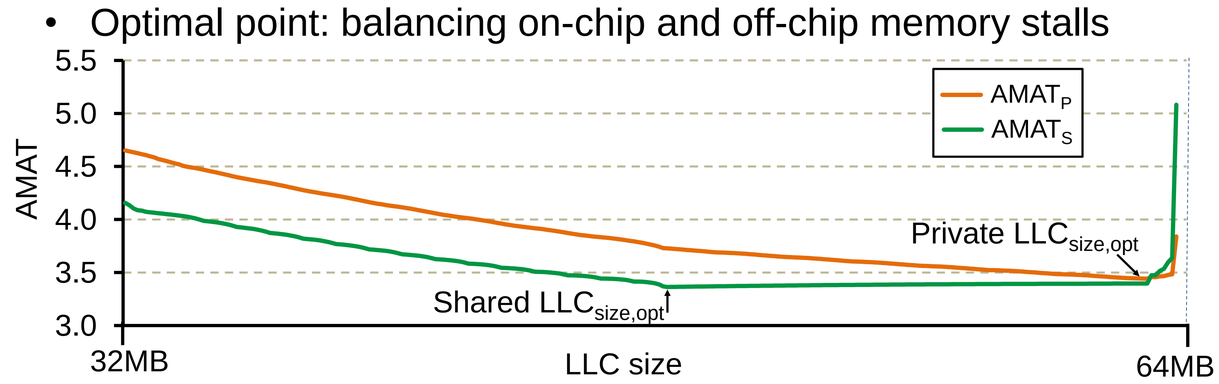


$$Ratio = \frac{AMAT_P}{AMAT_S}$$ <1.0 >1.0

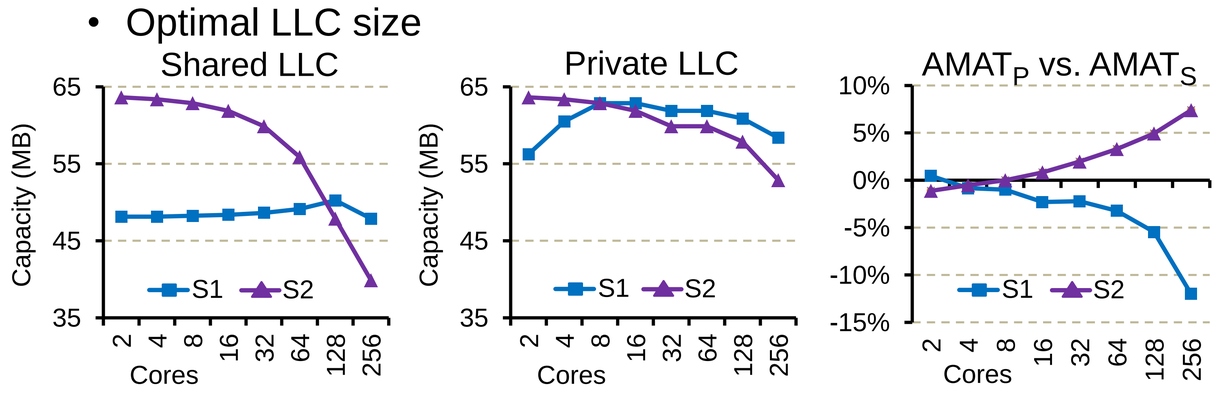


L2/LLC Capacity Partitioning

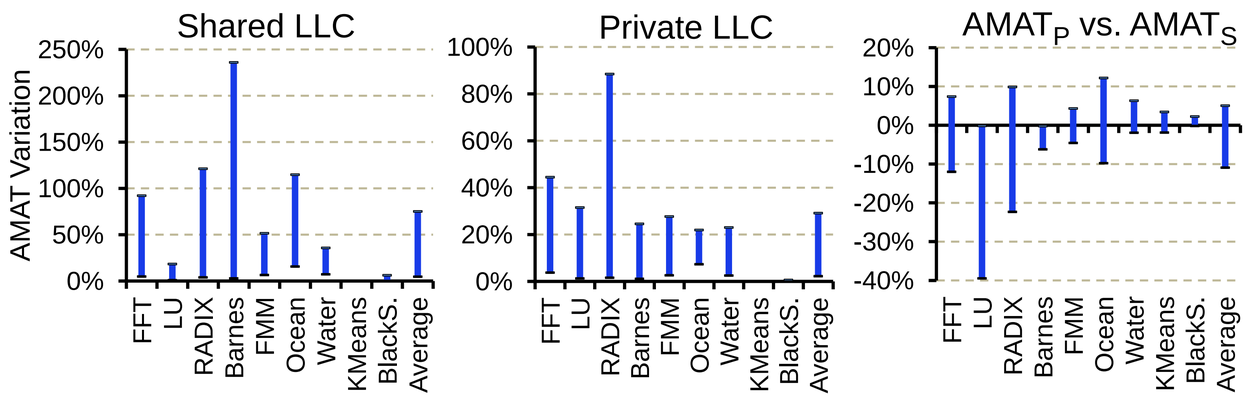
- L2 size + LLC size = 64MB
- Optimal point: balancing on-chip and off-chip memory stalls



- Optimal LLC size



- AMAT variation across design space
  - L2/LLC capacity partitioning > Private/Shared LLC selection



Conclusions

- Multicore RD analysis: fast, insightful, and complete picture
- Private/Shared LLC selection is complicated
- Optimal hierarchy: balancing on-chip and off-chip memory stalls
- L2/LLC capacity partitioning is crucial

Future Work

- Reconfigurable caches
- Dynamic memory management
- Evaluating applications' scalability