MichiganEngineering

Temporal Streams in Commercial Server Applications

Thomas Wenisch

Joint work with: Michael Ferdman[*], Anastasia Ailamaki [*†], Babak Falsafi [†], Andreas Moshovos[‡]

Advanced Computer Architecture Lab
University of Michigan

[*] Carnegie Mellon University

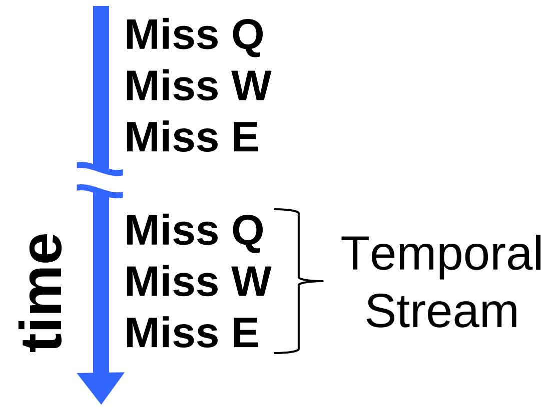
[†] Ecole Polytechnique Federale de Lausanne

[‡] University of Toronto

Temporal streams

Recurring off-chip load miss sequences

- Arise because data structure traversals repeat
- Orthogonal to strided streams
- 2 to 1000's of cache blocks long



Underlie numerous address-correlating prefetchers

E.g., [Chilimbi 02] [Solihin 02] [Wenisch 03] [Nesbit 04] [Ferdman 07] [Chou 07]

- Often effective when strides fail (e.g., pointer chasing)
- Bottom line (server apps): 40-60% coverage, 5-20% speedup

But, where is the miss repetition coming from?

This study's goals

Goal 1: Characterize streams independent of HW assumptions

- Challenge: Identify temporal streams in miss traces
- Approach: Use data compression to find repetition

Goal 2: Identify application behaviors that cause streams

- Challenge: Commercial apps are closed-source
- Approach: Tie streams to exported function names

Goal 3: Determine impact of memory system organization

- Contrast: CMP - Single-chip multiprocessor
DSM - Multi-chip distributed shared memory

Key insights

Most misses in long temporal streams

- 75% of misses in streams; median length ≈ 10 misses
- Justifies effectiveness of prior HW proposals

Coherence-intensive activities tend to be most repetitive

- Leads to drastic differences between DSM and CMP

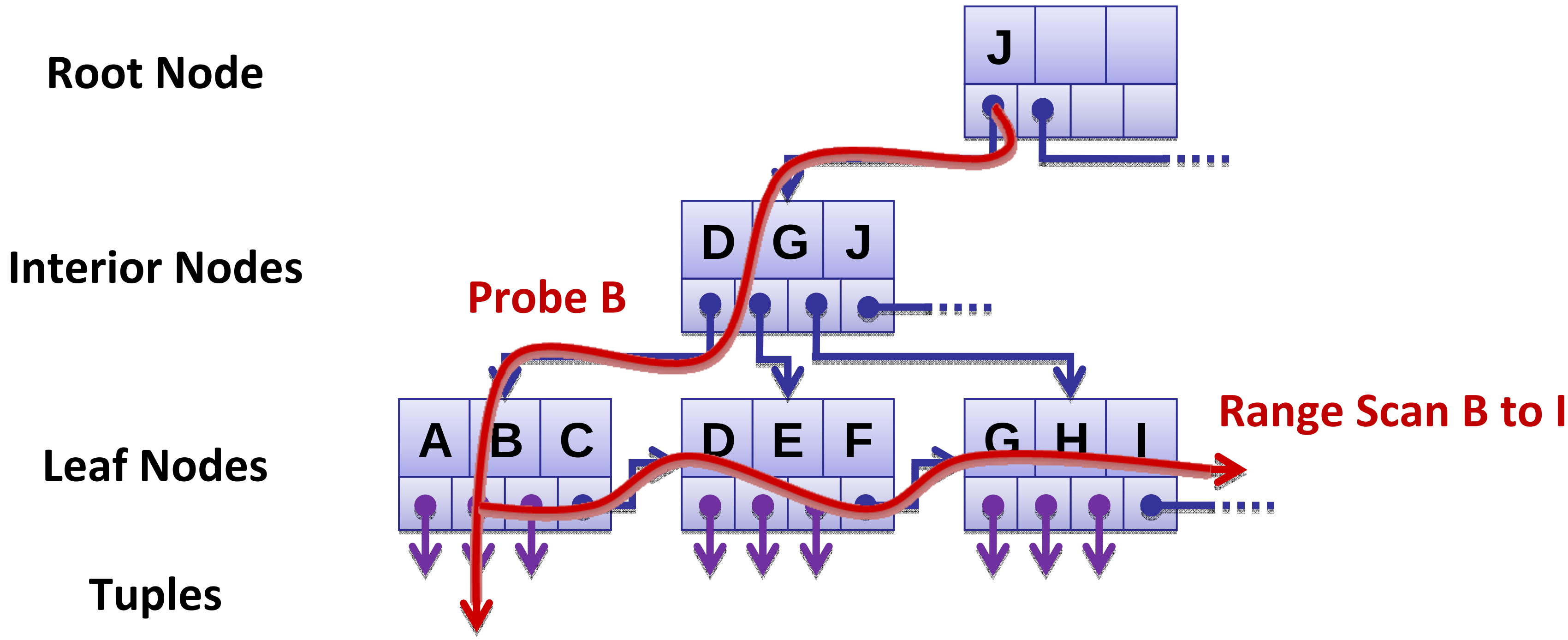
No single activity accounts for >20% of streams

- Commercial SW already highly-optimized

Outline

- Temporal stream examples from real SW
- Analysis methodology
- Results
  - Quantitative characterization
  - Code module analysis
- Conclusion

Example 1: B+Tree probes & range scans

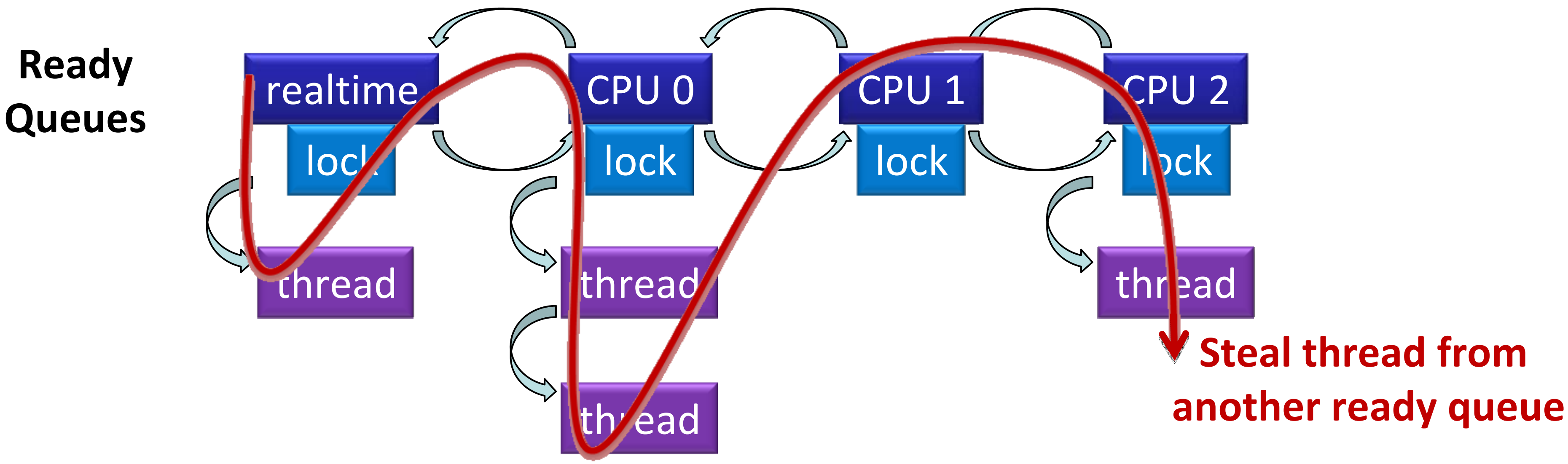


Temporal streams arise from…

- Repeated probes for the same key
- Overlapping range scans

Account for ~10% of temporal streams in OLTP

Example 2: Solaris kernel scheduler



Temporal streams arise because…

- CPUs steal threads when own ready queue is empty
- When stealing, all CPUs traverse queues in same order
- Locks & frequent queue updates → coherence misses

Account for ~10% of temporal streams on DSM

Analysis methodology

Step 1: Identify temporal streams

Analyze memory traces via data compression

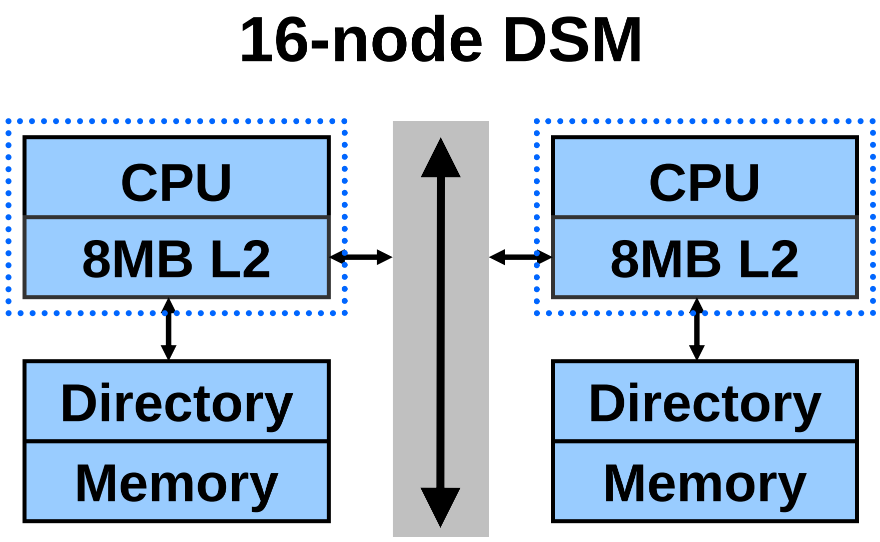
- SEQUITUR hierarchical compression *[Nevill-Manning 97]*
  - Heuristic for findings longest recurring sub-sequences
  - Used in prior control-flow and L1 access repetition studies *[Larus 99] [Chilimbi 02]*

Step 2: Connect to application functionality

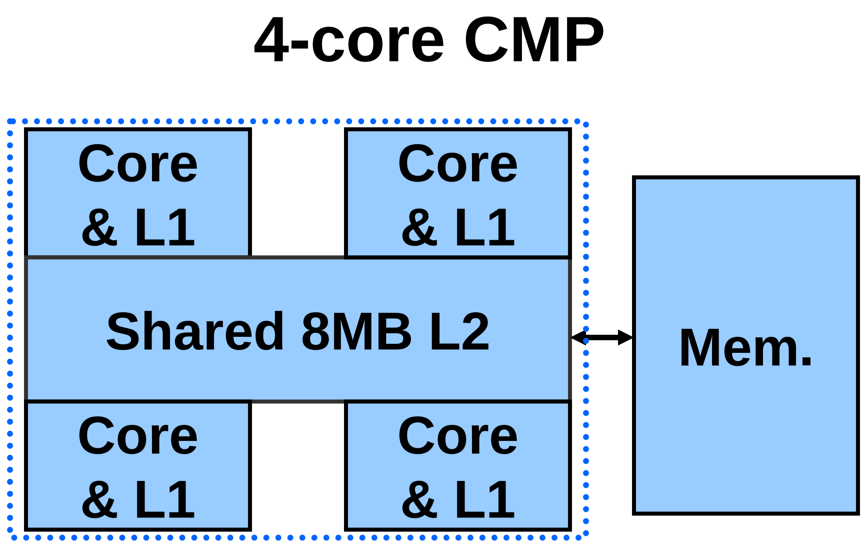
Deduce functionality from enclosing function names

- Function naming conventions aid categorization
  - Disclaimer: categorization based on educated guesses

System models & applications



Coherence misses



Capacity/conflict misses

Trace Collection

Full-system simulation with ***Flexus***

Off-chip (L2) read misses

Includes OS misses (Solaris 8)

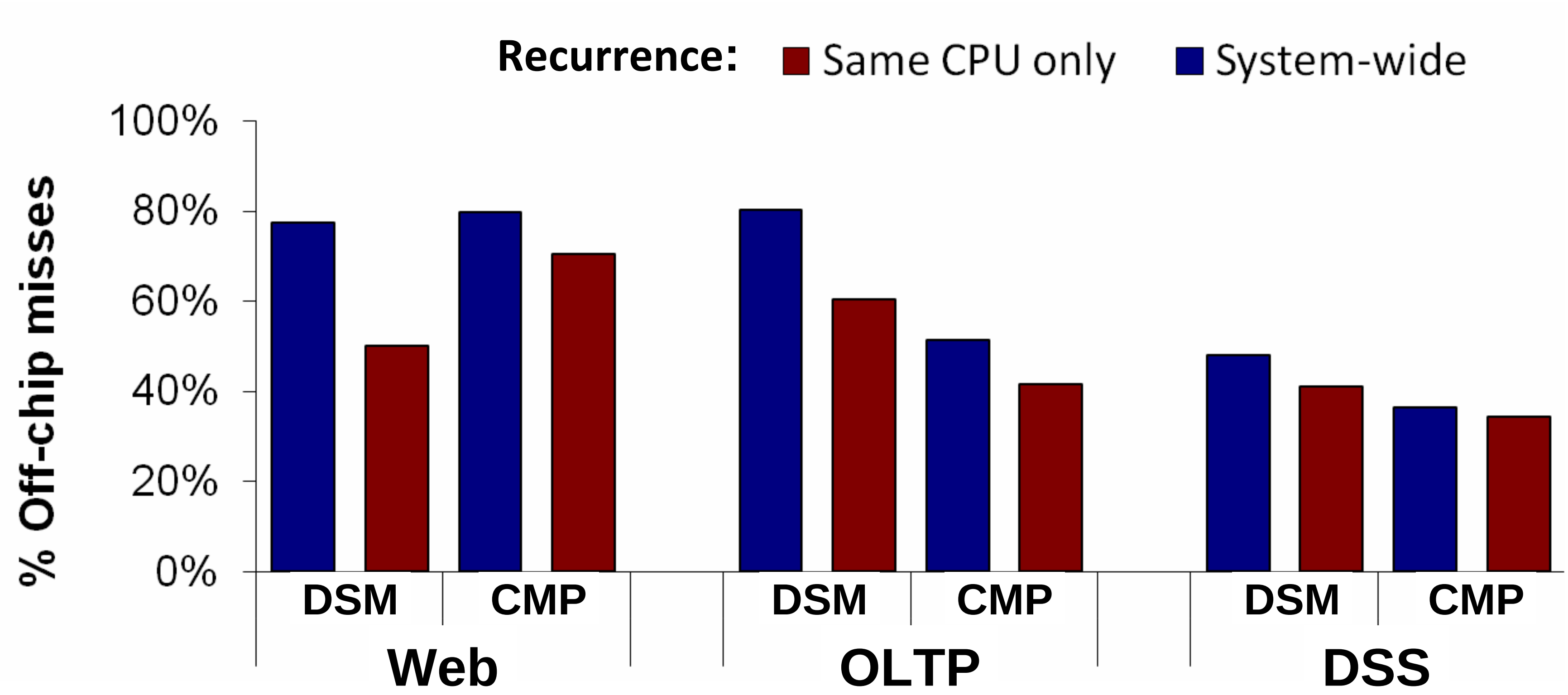
Applications

Web: SPECweb99 on Apache & Zeus

OLTP: TPC-C on DB2

DSS: TPC-H Queries 1, 2, 17 on DB2

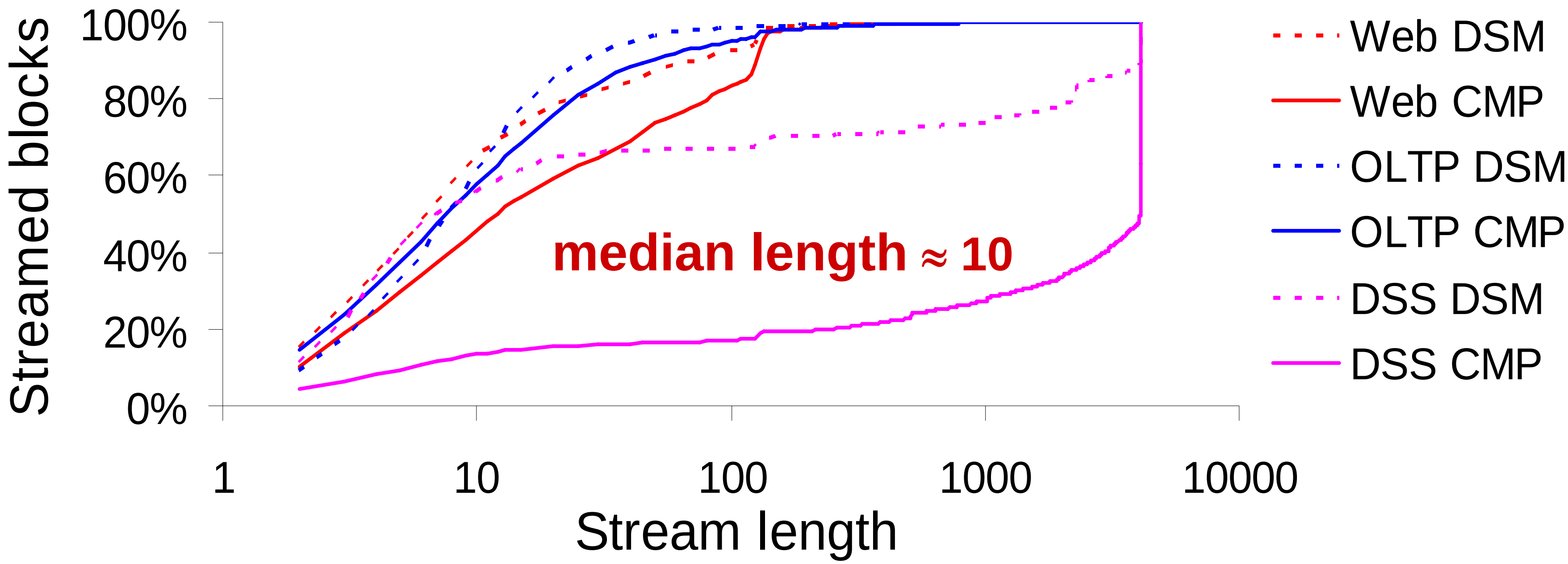
Fraction of misses in temporal streams



Avg. 75% misses in temporal streams

Streams recur across processors (esp. DSM)

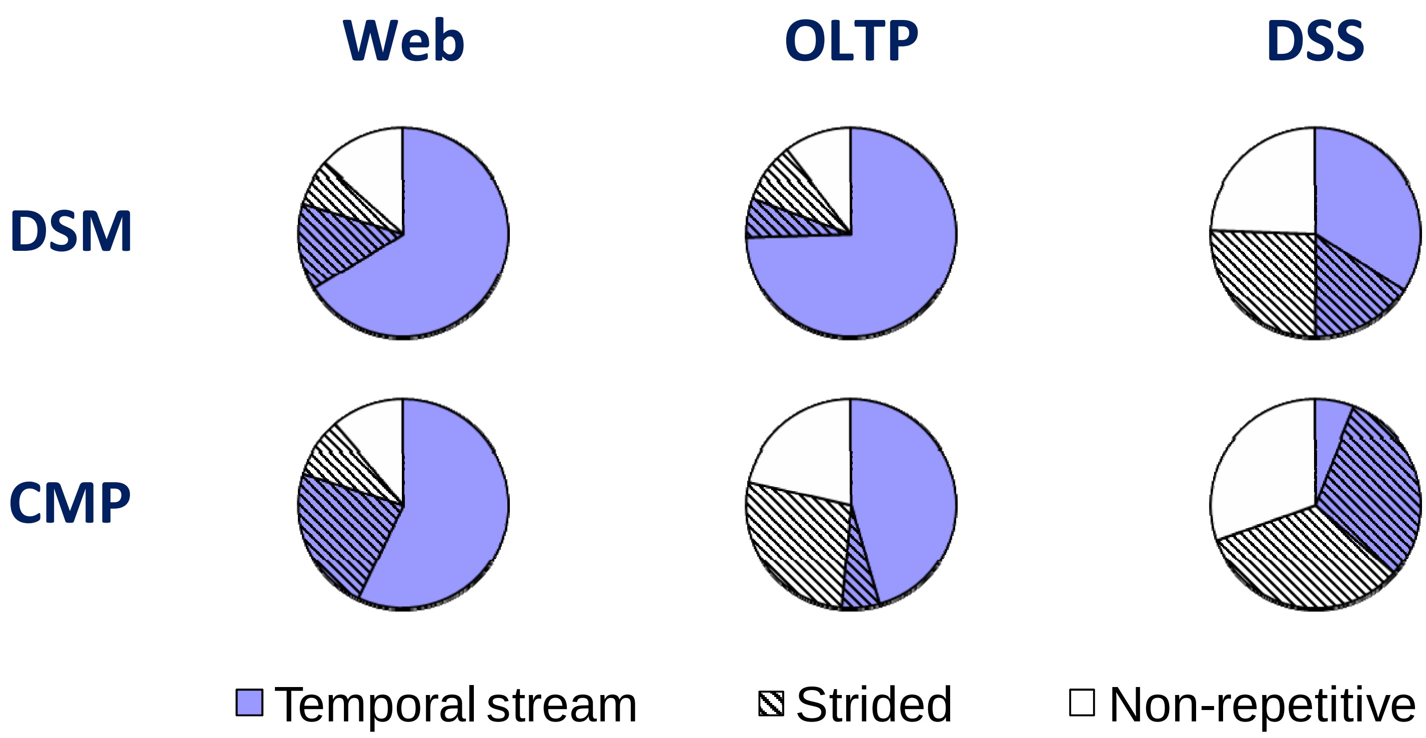
Temporal stream length



- Capacity misses tend to yield longer streams
- Above 512 cache blocks → typically bulk memory copies

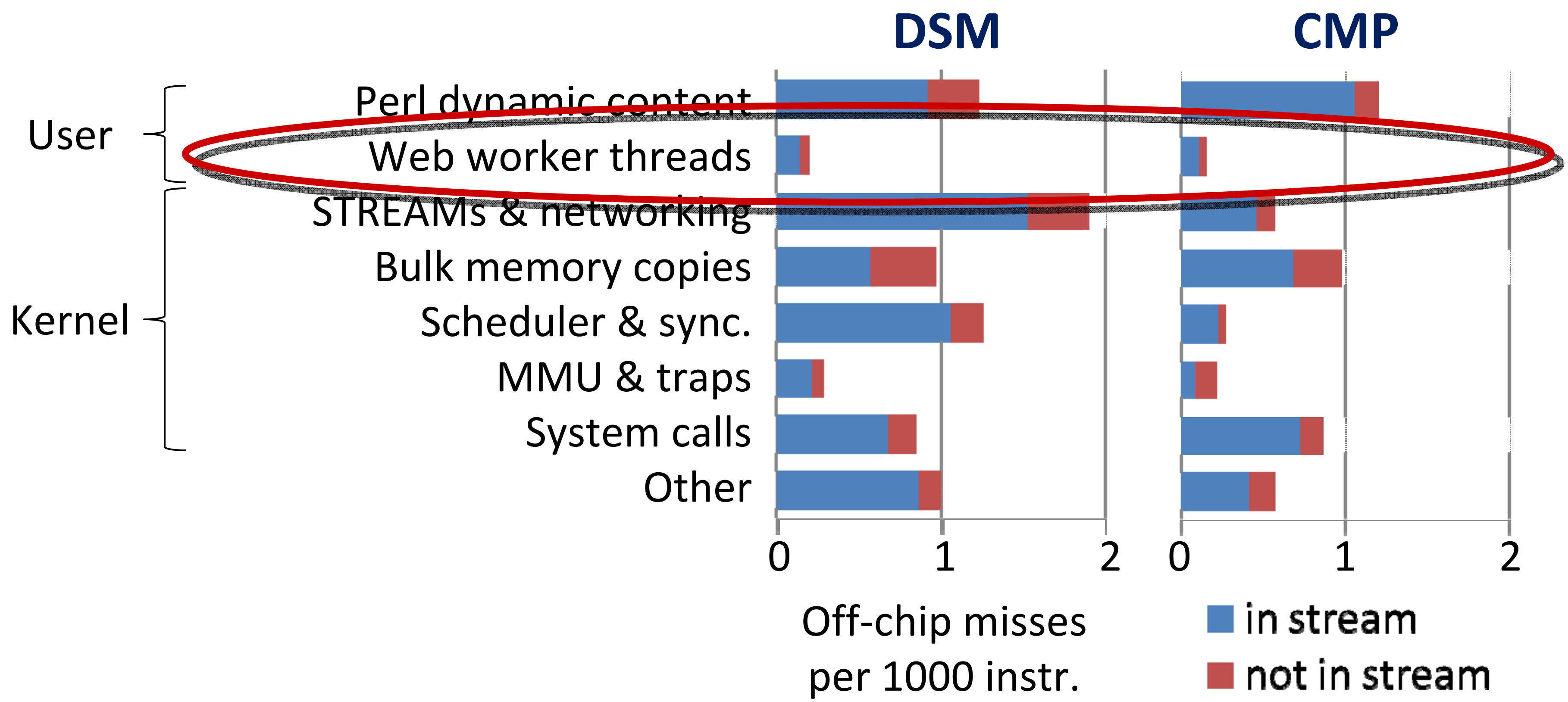
Long streams increase prefetching potential, but require more complex mechanisms (e.g., flow control)

Temporal streams vs. strided misses



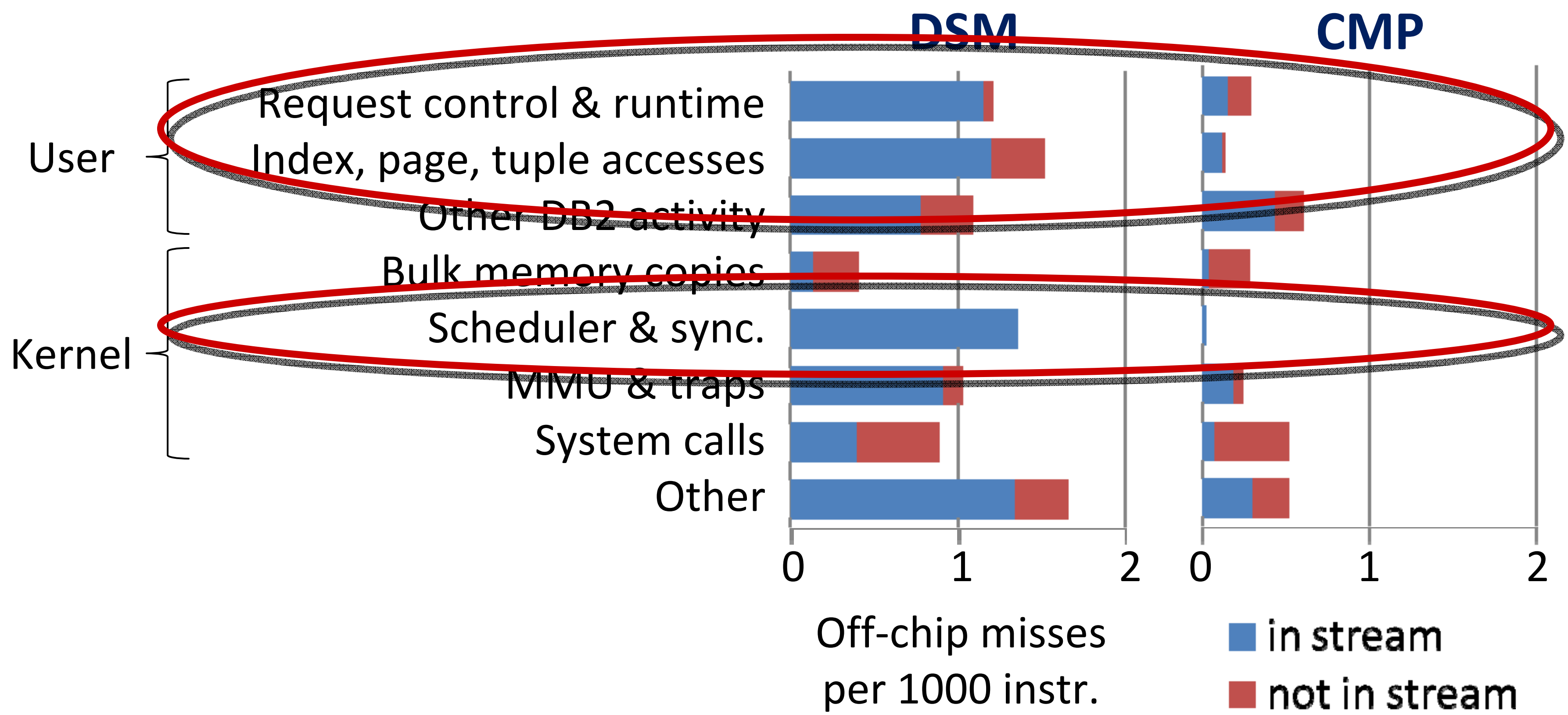
Temporal streams and strides target different accesses → coverage is largely disjoint

Stream sources: Web



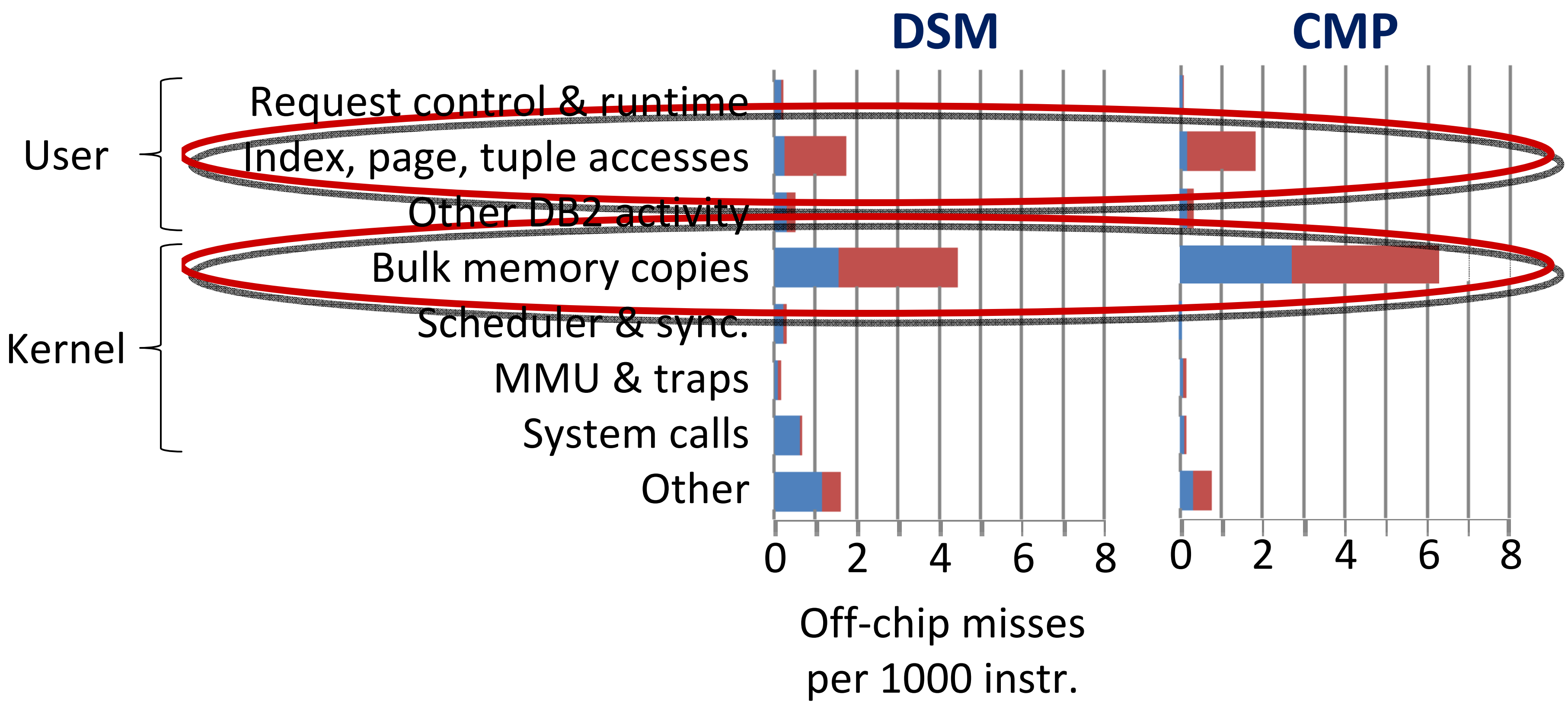
- Temporal streams pervasive across activities
- Web server incurs few misses; OS dominates

Stream sources: OLTP



- DSM: Lock/sharing-intensive activities highly repetitive
- CMP: intra-chip sharing – no off-chip coherence

Stream sources: DSS



- Non-repetitive bulk memory copies dominate
- Most data visited only once – few temporal streams

Conclusions

Temporal streams are…

- Pervasive: 75% of misses in streams
- Long: median length ≈ 10 cache misses
- Non-strided: synergistic with stride prefetching

Coherence and capacity misses behave differently →
alters temporal streams across CMP and DSM

Many critical OS functions yield temporal streams →
our results broadly applicable beyond Web, OLTP, DSS

Sponsors

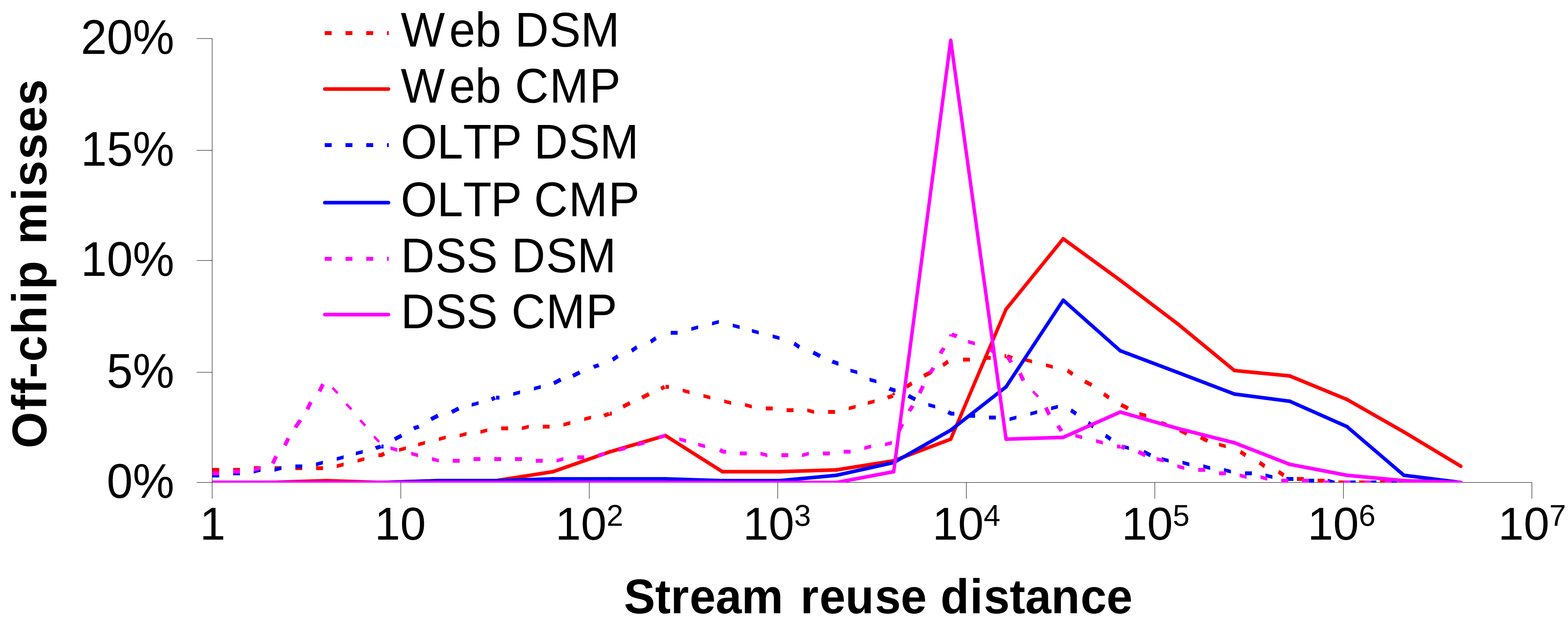
For more information

http://www.eecs.umich.edu/~twenisch

http://www.ece.cmu.edu/~stems

Backup

Stream reuse distance



- Coherence misses: reuse = *F*(sharing behavior) < 10^4 misses
- Capacity misses: reuse = *F*(L2 size) $\geq 10^4$ misses

Reuse distance $\geq$ ***10^5 misses → recent prefetchers store stream meta-data off-chip***

Related work: Prefetching

- Address correlation [Joseph 97] [Lai 01] [Solihin 02]
  - TMS extends pair-wise address correlation
  - TMS supports arbitrary sequence length
- Stride/spatial [Nesbit 04] [Sherwood 00] [Somogyi 06]
  - Can eliminate cold misses
  - TMS more effective for pointer-based structures
- Software-assisted [Chen 04] [Luk 99] [Roth 99]
  - Target pointer-based data structures
  - TMS parallelizes dependent misses