Database Servers on Chip Multiprocessors: Limitations and Opportunities

Nikos Hardavellas

With Ippokratis Pandis, Ryan Johnson, Naju Mancheril, Anastassia Ailamaki, Babak Falsafi

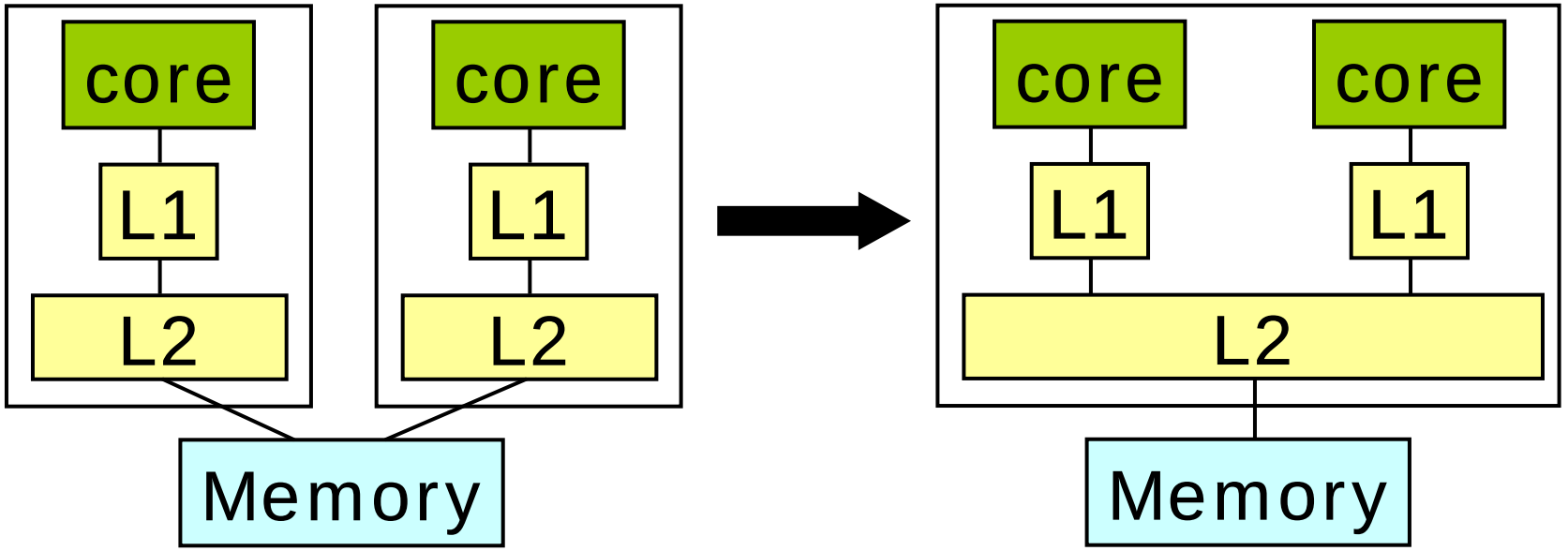
Databases @ Carnegie Mellon

CALCM

Hardware Integration Trends

traditional multiprocessors

chip multiprocessors



- Moore's Law: 2x trans. = 2x cores, 2x caches
- Trends to use larger but slower caches

Contributions

We show that:

1. L2 caches growing bigger and slower
   - Bottleneck shifts from Mem to L2
   - DBMS absolute performance drops

   Must enhance DBMS L1 locality

2. HW parallelism scales exponentially
   - DBMS cannot exploit parallelism under light load

   Need inherent DBMS parallelism

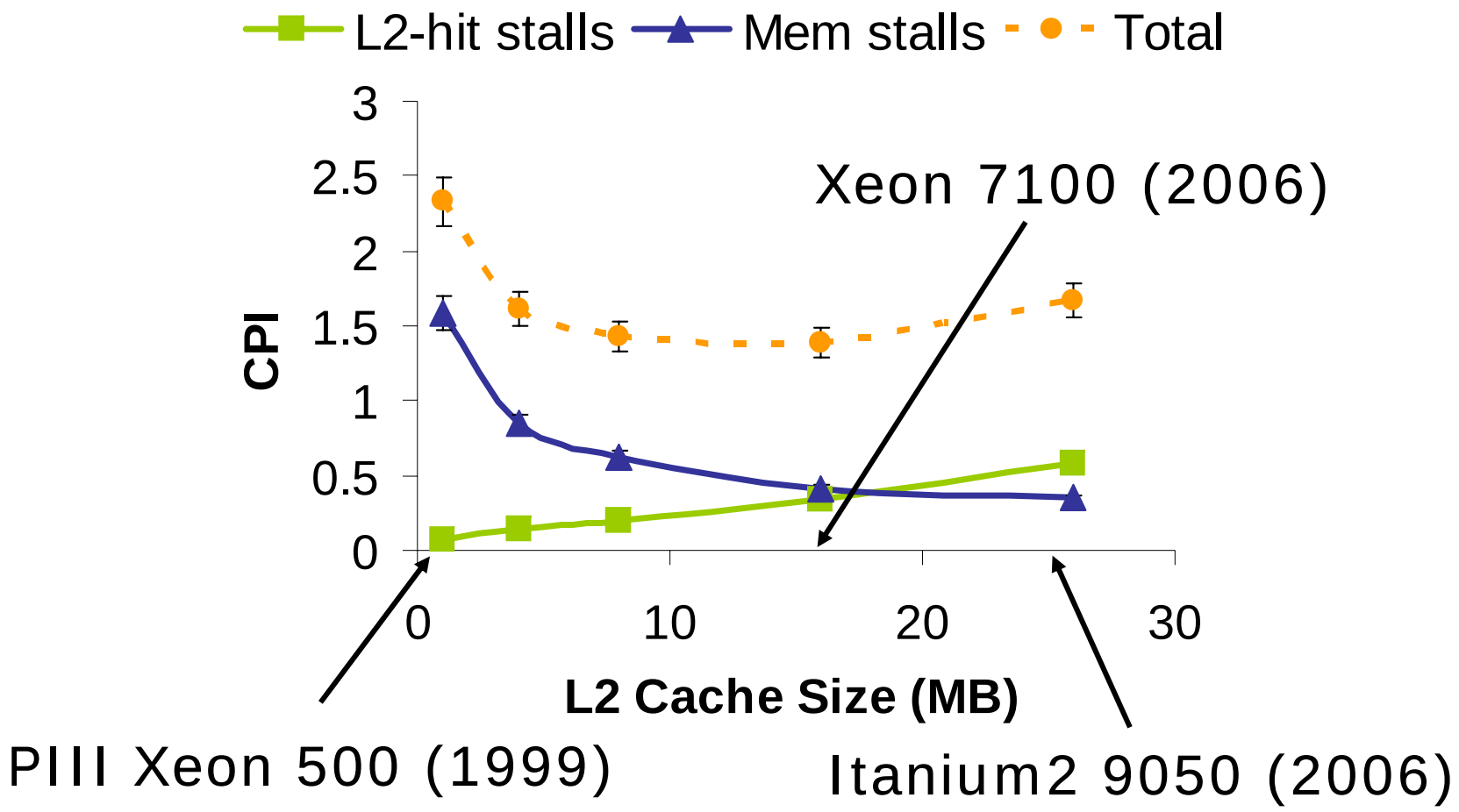
Methodology

- Flexus simulator (developed at CMU)
  - Cycle-accurate, full-system

- OLTP: TPC-C, 100wh, in memory
- DSS: TPC-H throughput, 1GB db, in memory
  - Scan- and join-bound queries (1, 6, 13, 16)

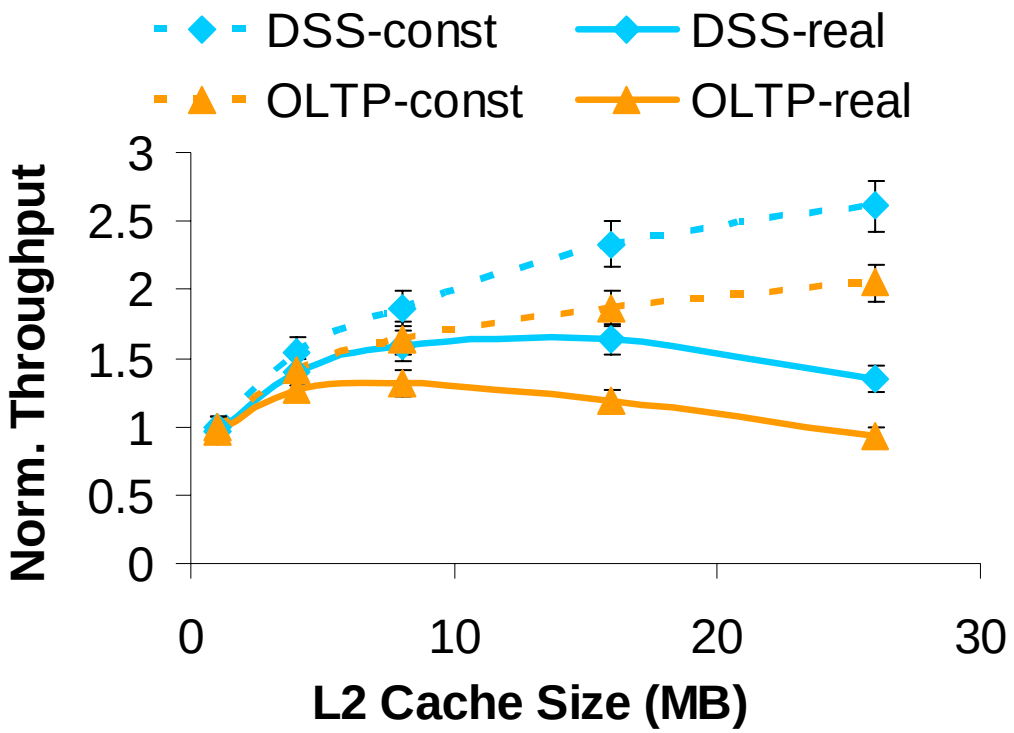
- Saturated: 64/16 clients (OLTP/DSS)
- Unsaturated (light load): 1 client

Observation #1
Bottleneck Shift to L2-hit Stalls



- Bottleneck shift from Mem stalls to L2-hit stalls

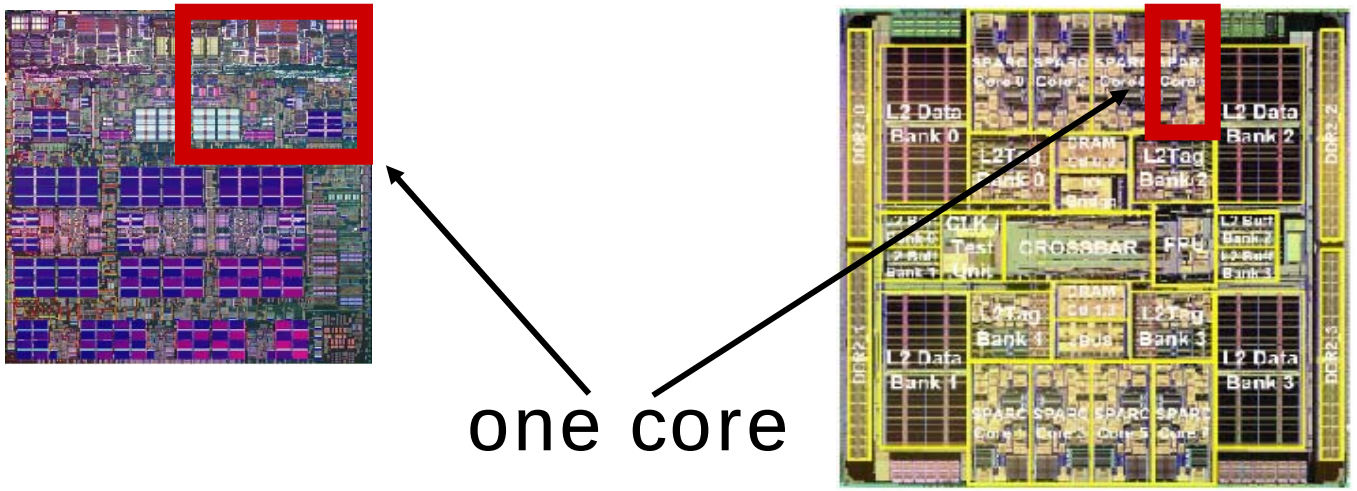
Impact of L2-hit Stalls



- Increasing cache size reduces throughput
- Must enhance L1 locality

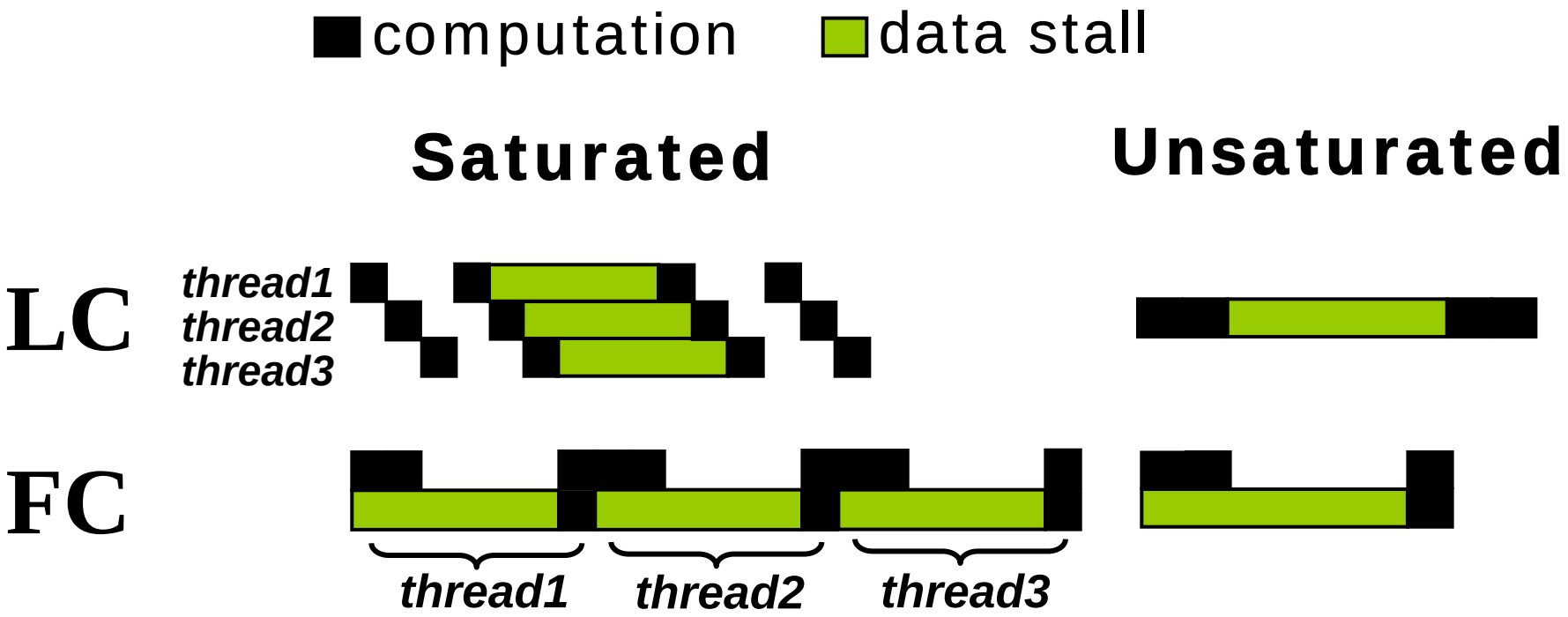
Observation #2
Parallelism in Modern CMPs

- **Fat Camp (FC)**
  wide-issue, OOO
  e.g., IBM Power5
- **Lean Camp (LC)**
  in-order, multi-threaded
  e.g., Sun UltraSparc T1



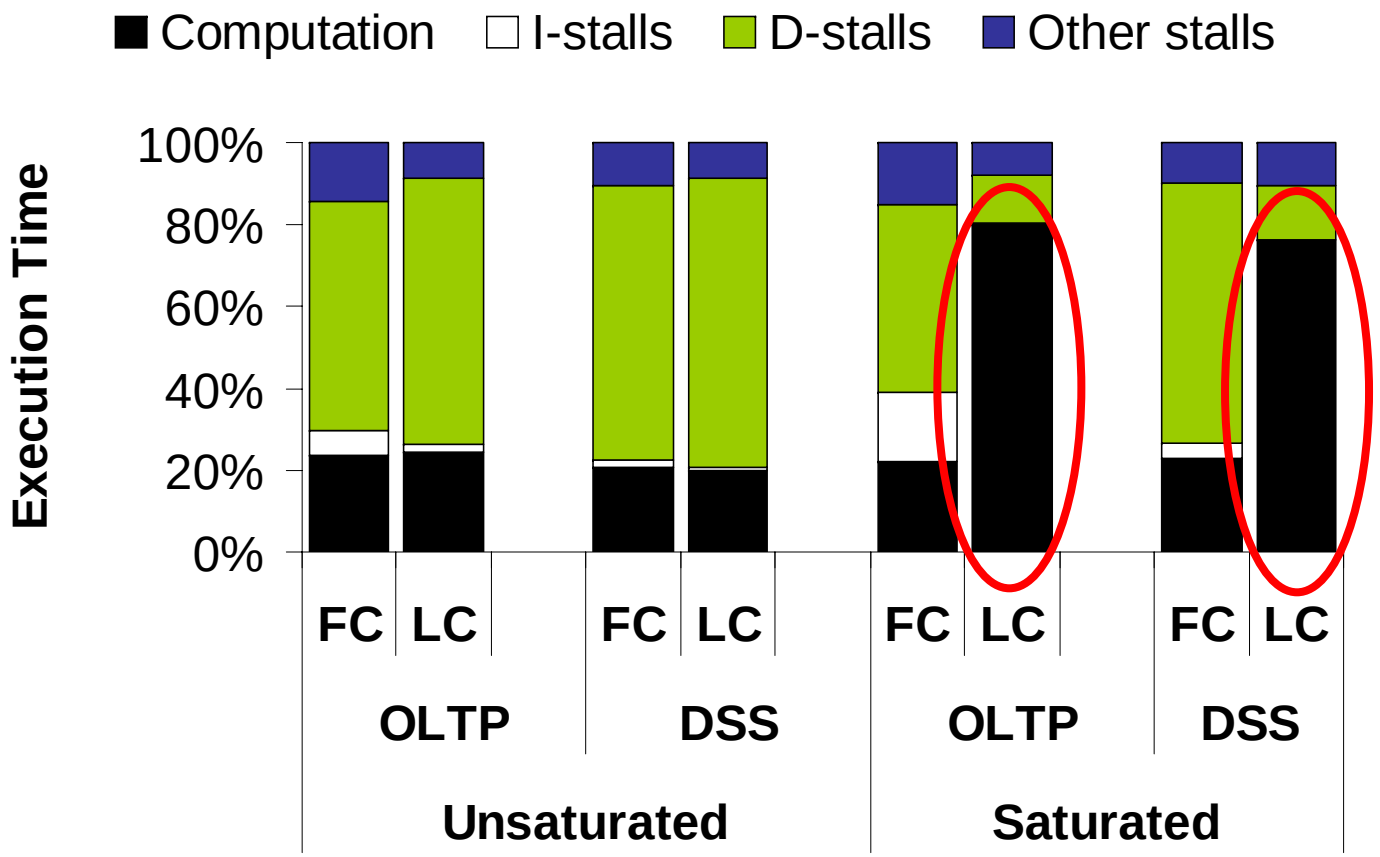
FC: parallelism within thread, LC: across threads

How Camps Address Stalls



- LC: stalls can dominate under unsaturated
- FC: stalls exposed in all cases

Prevalence of Data Stalls



- DBMS need parallelism & L1D locality

Impact

- L2 caches growing bigger and slower
- HW parallelism scales exponentially
  - Bottlenecks shift, data stalls are exposed
- DBMS must provide both
  - Fine-grain parallelism across and within queries
  - L1 locality

http://www.cs.cmu.edu/~stageddb/