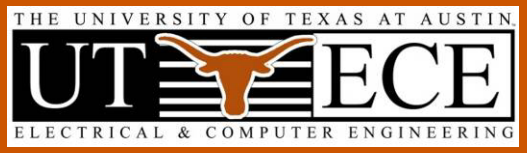


ISCA 2010

The Virtual Write Queue: Coordinating DRAM and Last-Level Cache Policies

Jeffrey Stuecheli[1,2], Dimitris Kaseridis[1], David Daly[3], Hillery C. Hunter[3] & Lizy K. John[1]

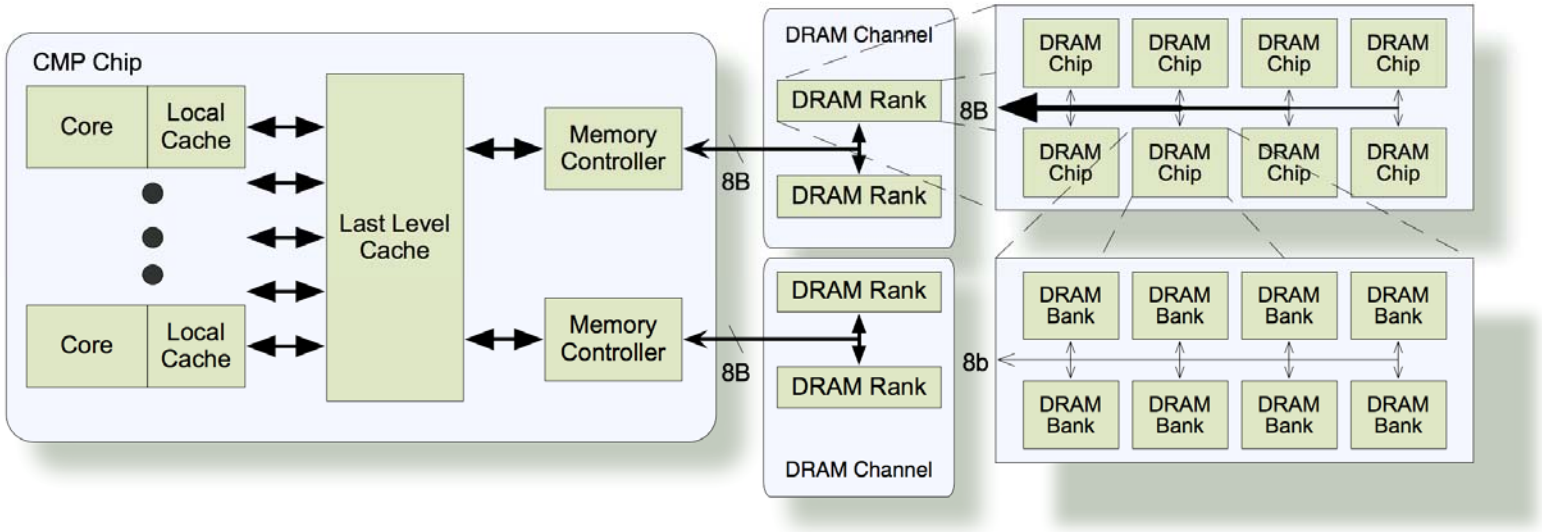
[1]ECE Department, The University of Texas at Austin

[2]IBM Corp., Austin

[3]IBM Thomas J. Watson Research Center

Memory terminology

- **Target System: Multi-Core CMP**
  - 8-16 cores (and up)
  - Shared cache and memory subsystem
- **Terminology:**
  - Channel/Rank/Chip/Bank
- **Area of focus: Improving scheduling of memory interface in light of many cores combined with DRAM technology challenges**

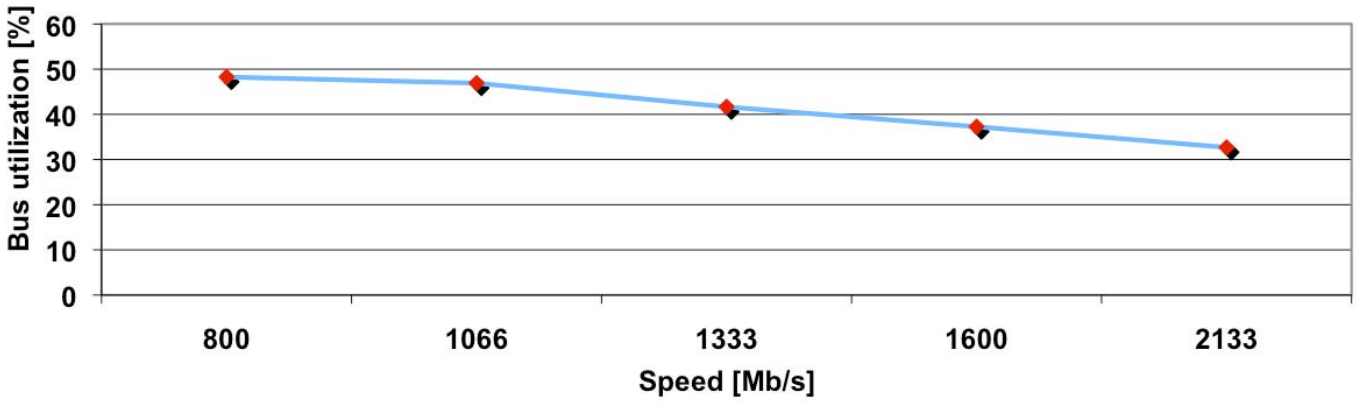


Memory Wall (Labyrinth)

- **Traditional concern is read latency**
  - Fixed at ~26 ns
- **Beyond latency, many parameters are limiters to efficient utilization**
- **Data bus frequency →2x each DDRx generation**
  - DDR 200-400, DDR2 400-1066, DDR3 800-1600
  - But, internal latency is ~constant
- **Fixed latency**
  - Bank Precharge (50ns, ~7 operations@1066Mhz)
  - Write→Read (7.5ns, ~2 operations@1066MHz)

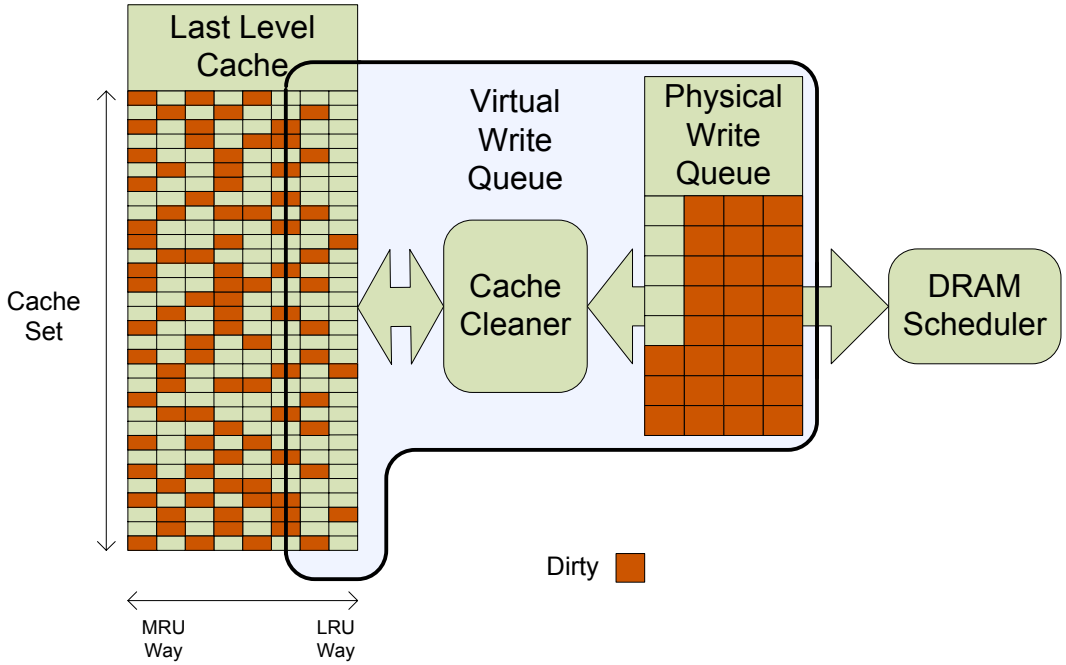
Implications

- **Scheduling efficiency**
  - Reads → Critical path to execution
  - Writes → Decoupled
- **Queuing**
  - We need more write buffering (make the most of each opportunity to execute writes)
  - Not Read buffering due to latency criticality of loads



The Virtual Write Queue

- **Grow effective write reordering by an order of magnitude through a two-level structure**
  - Writes can only *execute* out of physical write queue
  - Keep physical queue full with a *good* mix of operations
  - Physical write queue becomes staging ground, covers latency to pull data from the LLC.



VWQ Details

Cache→Memory Writeback Evolution

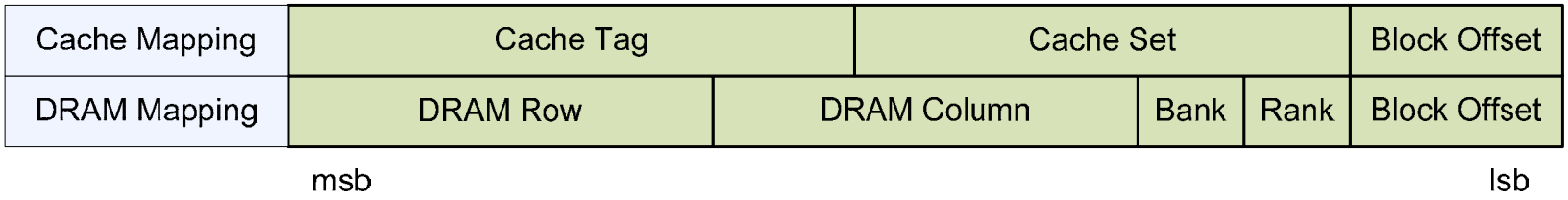
- **Forced Writeback:** Traditional approach to writeback.
- **Eager Writeback:** Decouple cache fill from writeback with early "eager" writeback of dirty data (Lee, MICRO 2000).
- **Scheduled Writeback:** Our proposal. Place writeback under the control of the memory scheduler.

Filling the Physical Write Queue

- **Key concept:**
  - Relatively few classes of writes:
    - Rank Classification: Which Rank?
    - Page Mode: Quality level
    - Bank conflicts: Avoid writes to same bank, different page
  - Physical Write Queue Content:
    - Maintain high quality writes in structure
    - Keep Writes to each Rank

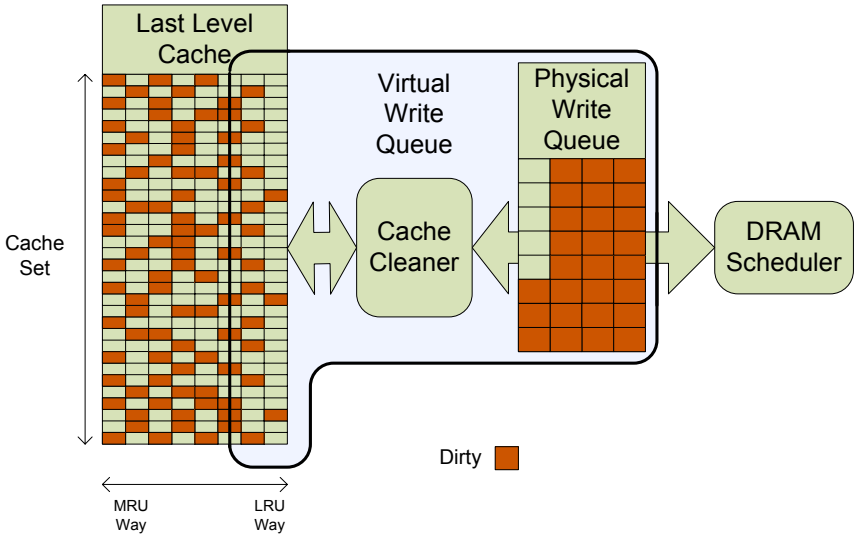
Address Mapping

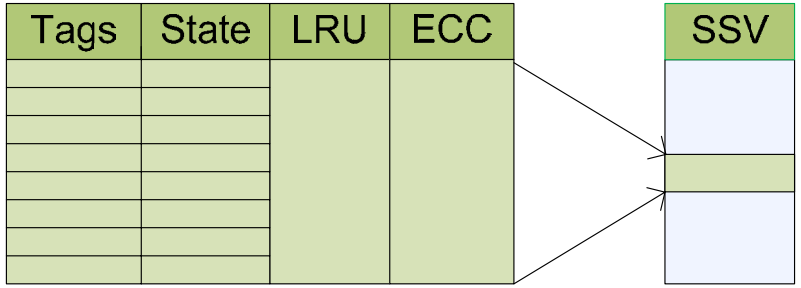
- **Set address of cache contains**
  - All Rank selection bits
  - All Bank selection bits
  - Some number of Column bits (address within a DRAM page)



The Cache Cleaner

- **Goal: fast/efficient search of large LLC directory**
- **Based around Set State Vector (SSV)**
- **SSV enables**
  - Efficient communication of dirty lines to be cleaned
- **Cleaner will select line based on current physical write Q contents**
  - Keep full with uniform mix of operations to each DRAM resource





Set State Vector

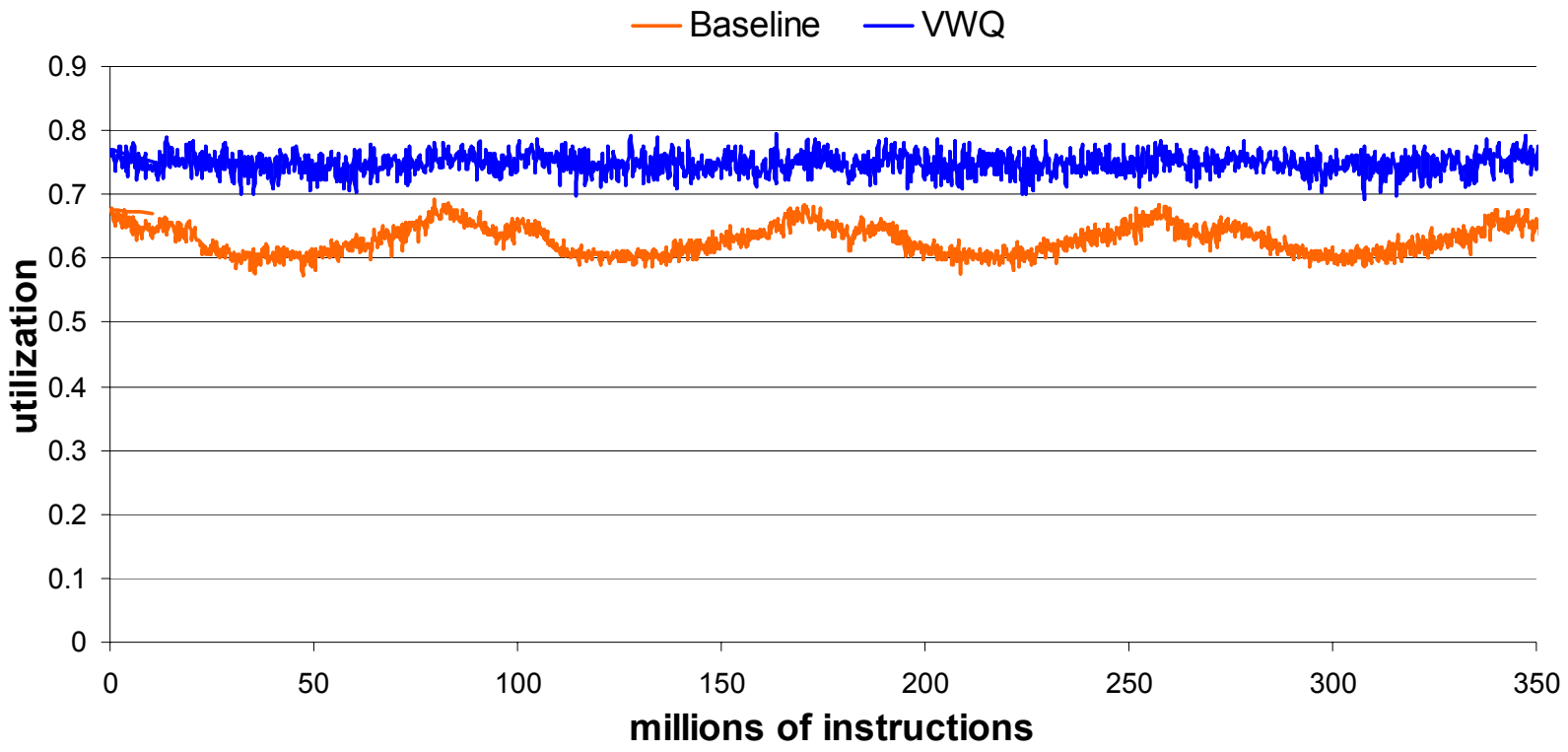
Read/Write Priority in scheduler

- Goal: **Defer write operations as long as possible**
  - *Forced Writeback*: Queuing depth is quite limited.
  - *Eager Writeback*: Write queue is always full; how do we know when we must execute writes?
  - *Virtual Write Queue*: Monitor overall fullness on a per Rank basis. Much larger effective buffering capability.

Evaluation/Results

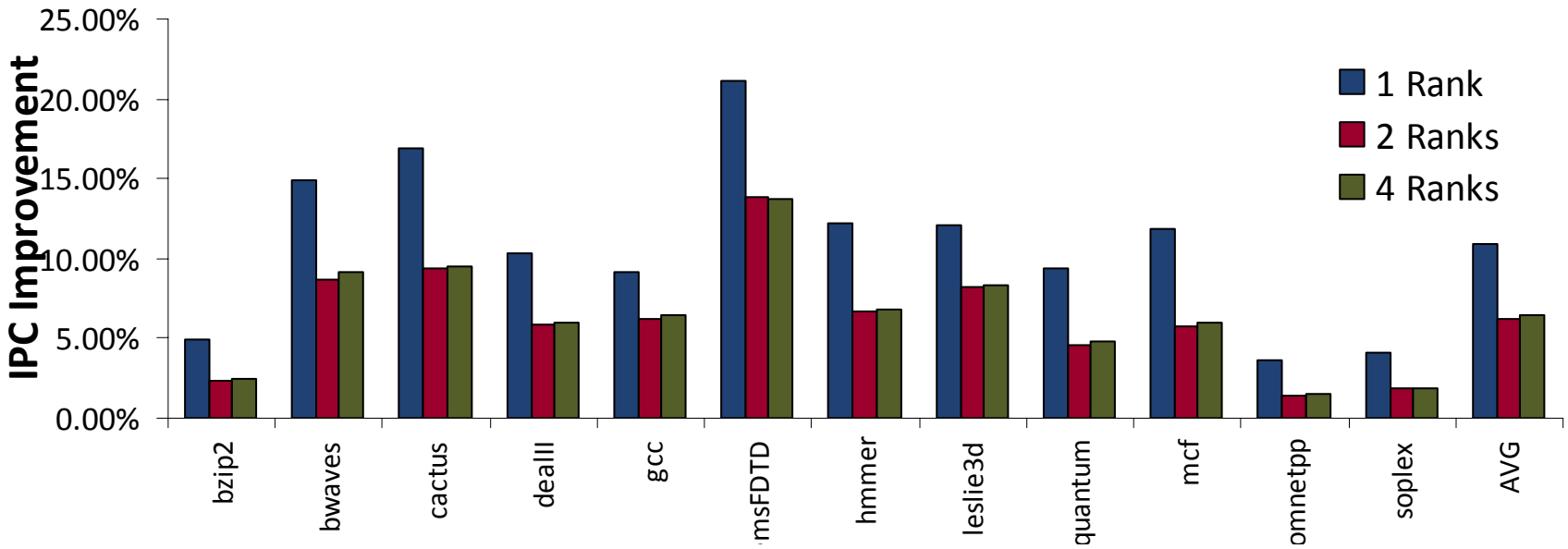
Bandwidth Improvement Example

- **From SPEC mcf workload**



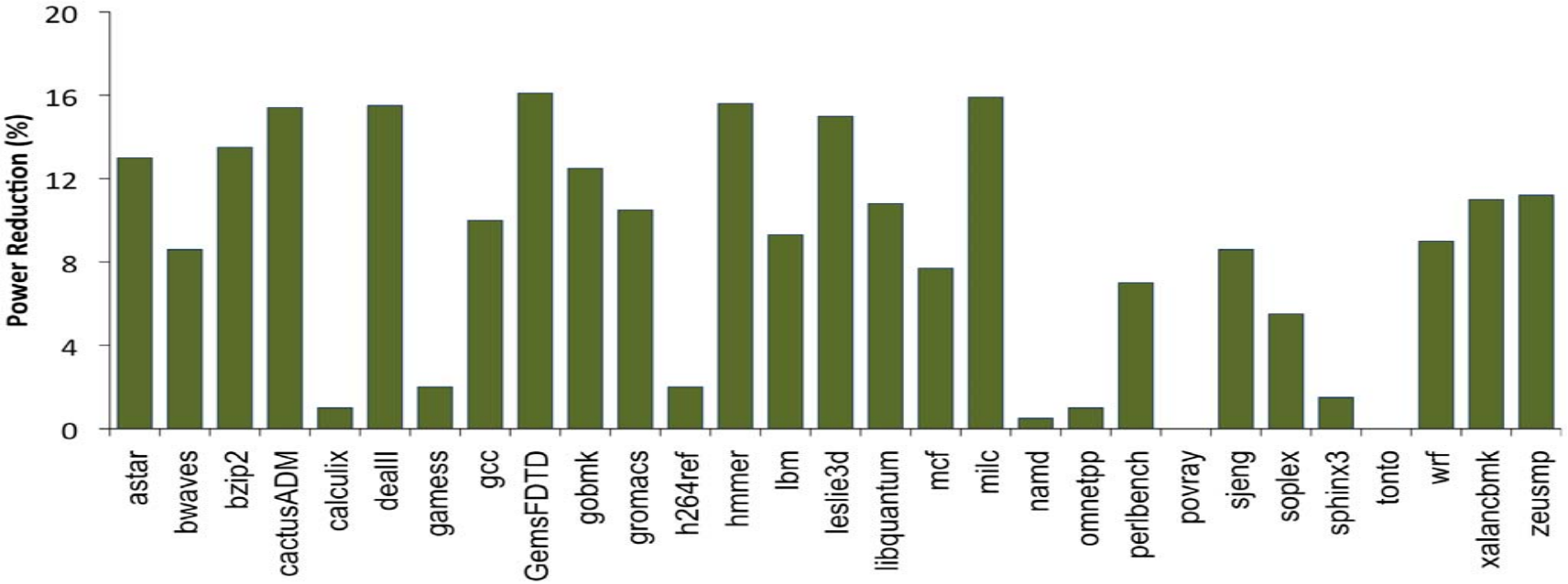
Virtual Write Queue IPC Gains

- **Each experiment consists of 8 copies of the same benchmark**
  - IPC was observed to be uniform across cores (symmetrical system was fair)
- **Improvements in 1,2, and 4 rank systems**
  - Largest improvement with 1 rank due to exposed “Write to Read Same Rank” penalty



Power Reduction Due to Increased Write Page Mode Access

- **Overall DRAM power reduction is shown**



Conclusion

- **Memory scheduling is critical to CMP design**
- **We must leverage all state in the SOC/CMP**

Thank You, Questions?

Laboratory for Computer Architecture
University of Texas Austin
&
IBM Austin
&
IBM T. J. Watson Lab