

*Presented at the 1998 HPCA Workshop on Computer Architecture
Evaluation Using Commercial Workloads*

System Design Considerations for a Commercial Application Environment

Luiz André Barroso and Kourosch Gharachorloo
Western Research Laboratory

Contributions from: Edouard Bugnion, Jack Lo, and Parthas Ranganathan

Myths About Database Applications

“I/O performance is all that really matters”

“Most of the time is spent in the OS”

“You need a \$1M+ system to study it”

“Applications are too complex for a simulation environment”

Shift in Bottleneck

- DB applications used to be I/O bound
- These days I/O performance matters, but...
 - I/O bandwidth/latency/architecture has improved
 - DB engine software has evolved to tolerate I/O latency
- Most applications can become CPU bound
- Today's challenges:
 - **Memory system!**
 - Processor architecture

Outline

- Introduction
- Scaling down DB workloads
- Tools and methods
- Memory system performance
- Processor architecture
- Summary

Scaling Down DB Workloads

Do you want to predict TPC numbers or study computer architecture?

Scaling Down DB Workloads

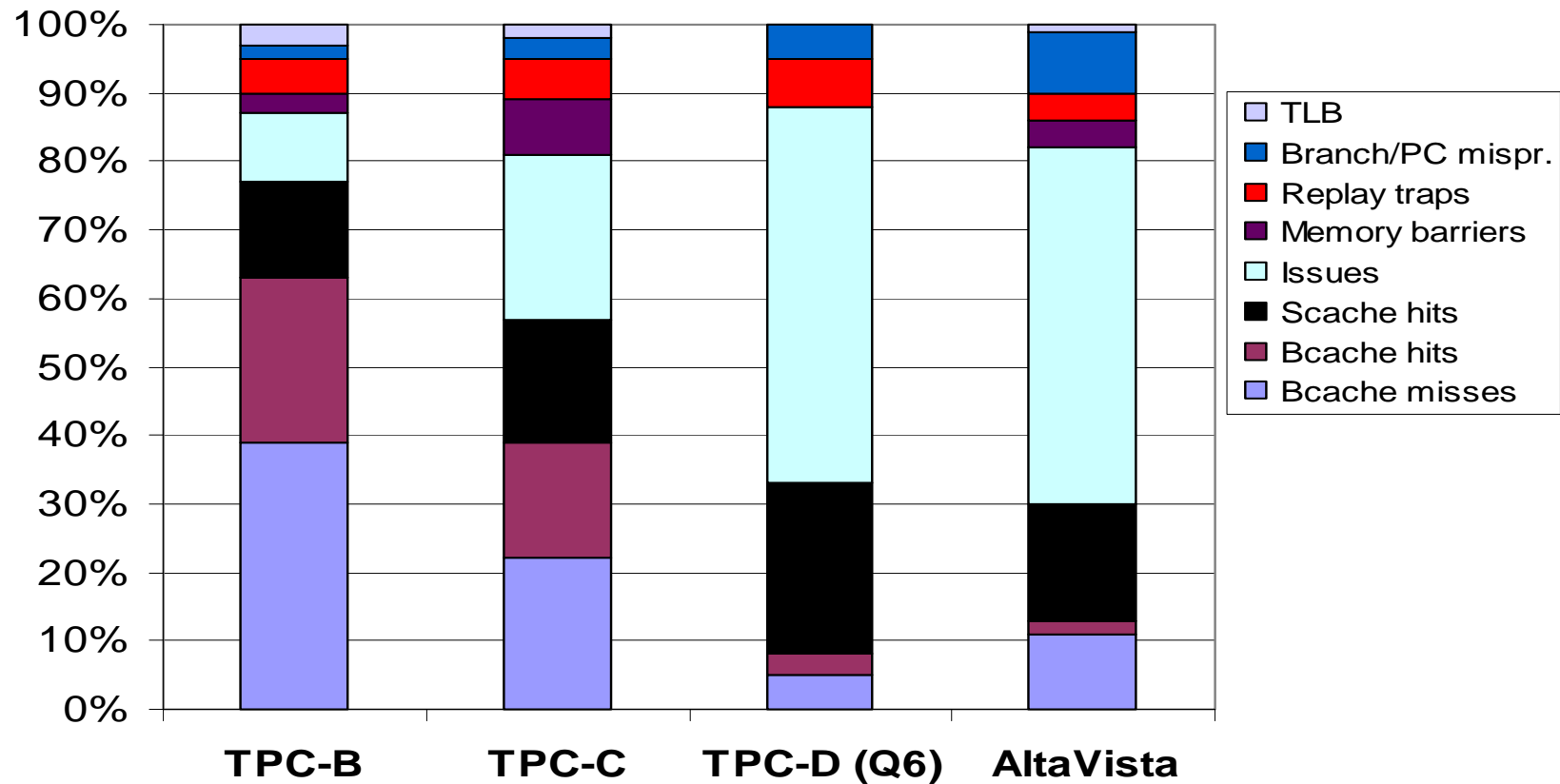
- Not a trivial task, but a feasible one
 - Requires deep understanding of the workload behavior
 - Depends on the target of the study
 - Critical for enabling simulation studies
- Extensive monitoring of the native application is required before simulation
- In-memory runs can be used for memory system studies
- Important to exclude idle time from calculated statistics

Tools and Methods

- Good hardware event counters (as in the 21164) are key
- Rich set of performance tools (ATOM, DCPI, IPROBE) enable:
 - Careful tuning and detailed profiling of the code
 - Detailed breakdown of the causes of processor stalls
 - Validation of scaling assumptions
- Powerful simulation infrastructure (SimOS-Alpha)
 - Account for both user and system behavior
 - Study complex applications “out-of-the-box”
 - Access to events not visible in a running system
 - Enable the study of future designs

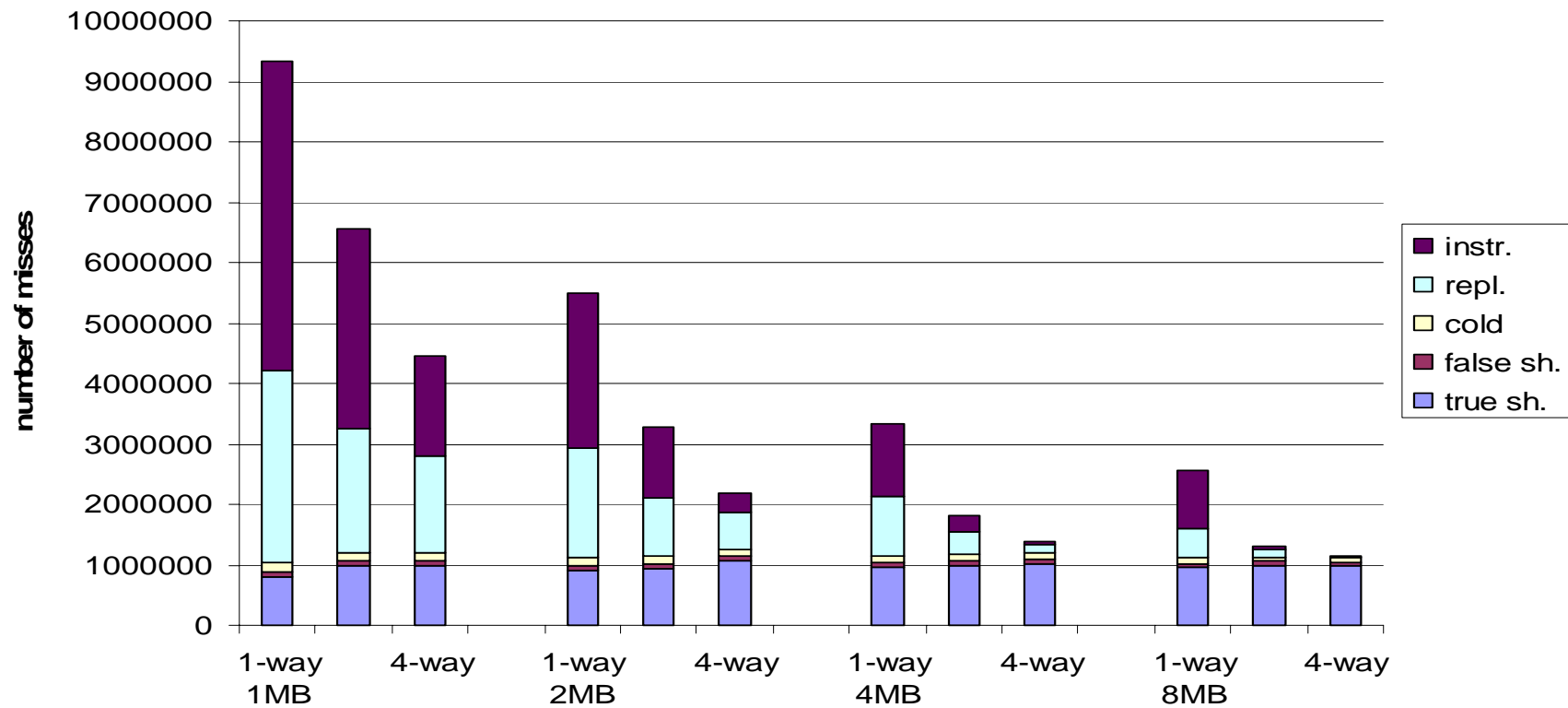
Memory System Performance

Breakdown of CPU cycles



OLTP: Effect of Cache Organization

SimOS: L2 miss landscape vs. cache configuration (P=4)



OLTP: DB Server Data Access Patterns

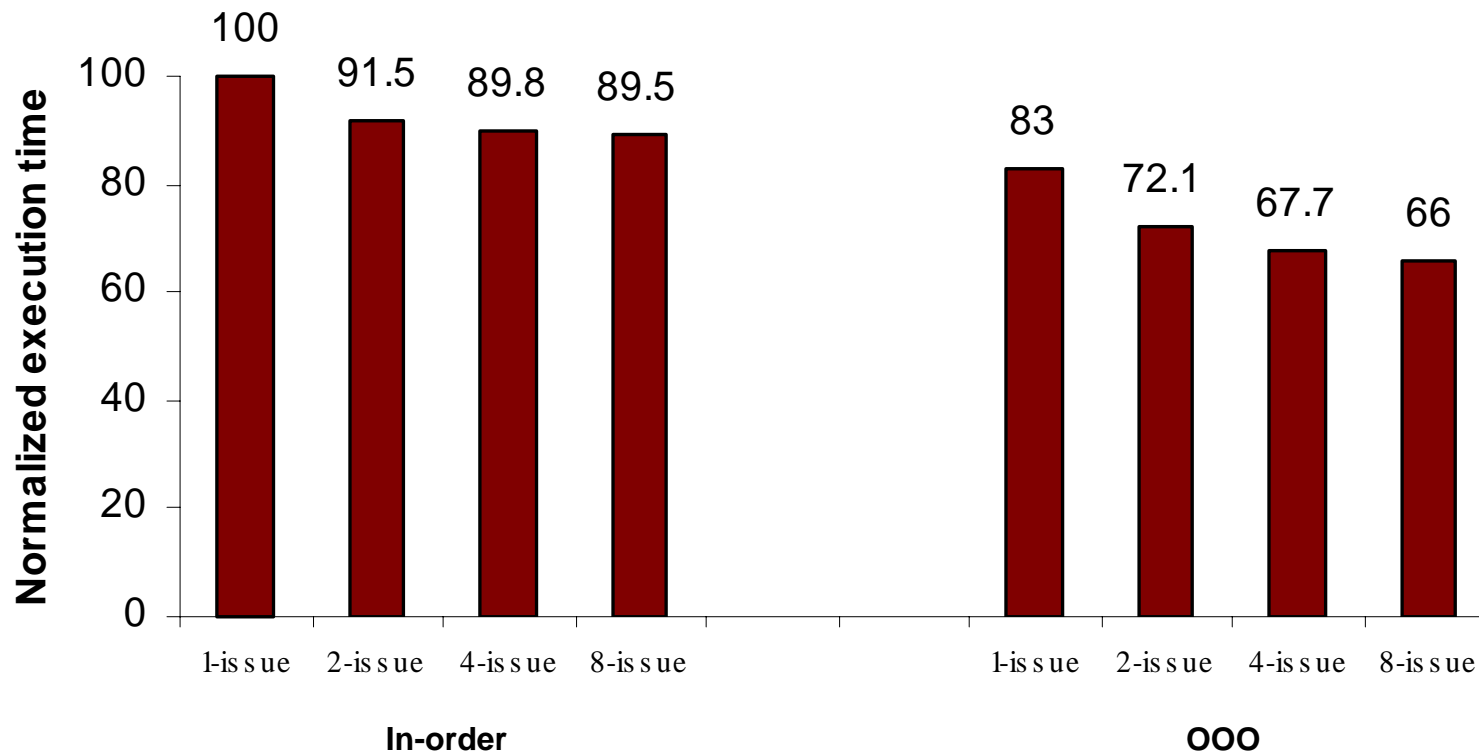
	Accesses	Bcache misses	Dirty misses*
Private data	75.3 %	10.3 %	0.0 %
Shared data (metadata)	21.8 %	80.4 %	95.3%
Shared data (block buffer)	2.9 %	9.3 %	4.7 %

P=4, 8MB Bcache

* over 60% of Bcache misses are dirty

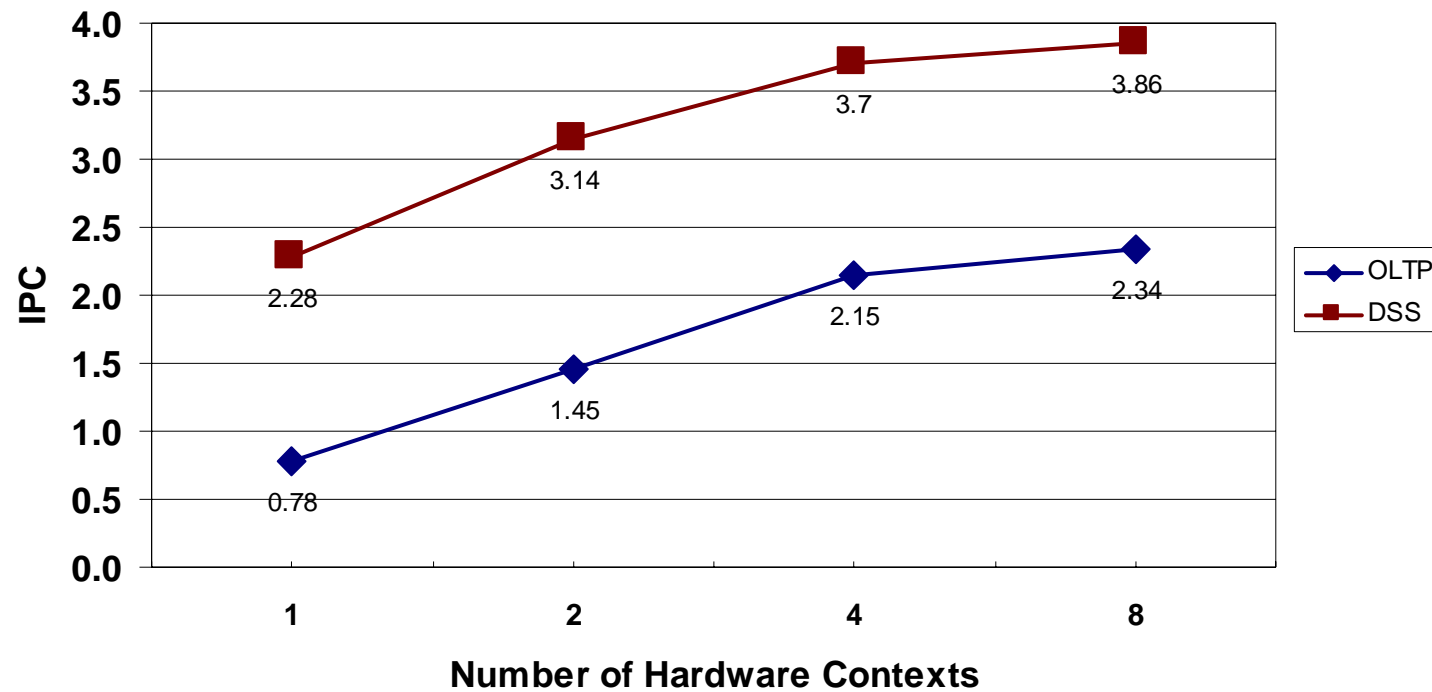
Processor Architecture

OLTP: impact of issue width and OOO issue



Processor Architecture

Simultaneous Multithreading



Summary

- Memory system is the current challenge in DB performance
- Problem size can be scaled down, but very carefully
- Combination of monitoring and simulation is very powerful

- Diverging memory system designs:
 - OLTP: use large/fast secondary caches and optimize the dirty miss case
 - DSS, AltaVista: use large/fast on-chip caches; may perform better without a secondary cache
- Both out-of-order and wider issue (up to 4-way) help
- SMT could improve OLTP performance significantly