DBMS on a modern processor: where does time go?

Anastasia Ailamaki, David DeWitt, Mark Hill and David Wood
University of Wisconsin-Madison

Presented by:
Bogdan Simion
Current DBMS Performance
Where is query execution time spent?

Identify performance bottlenecks in CPU and memory
Outline

• Motivation
• Background
• Query execution time breakdown
• Experimental results and discussions
• Conclusions
Hardware performance standards

- Processors are designed and evaluated with simple programs

- Benchmarks: SPEC, LINPACK

- What about DBMSs?
DBMS bottlenecks

• Initially, bottleneck was I/O
• Nowadays - memory and compute intensive apps
• Modern platforms:
  – sophisticated execution hardware
  – fast, non-blocking caches and
• Still ...
  – DBMS hardware behaviour is suboptimal compared to scientific workloads
Stalls overlapped with useful work !!!
Execution time breakdown

\[ T_Q = T_C + T_M + T_B + T_R - T_{OVL} \]

- \( T_C \) - Computation
- \( T_M \) - Memory stalls \( \{ \text{L1D, L1I}, \text{L2D, L2I}, \text{DTLB, ITLB} \} \)
- \( T_B \) - Branch Mispredictions
- \( T_R \) - Stalls on Execution Resources \( \{ \text{Functional Units}, \text{Dependency Stalls} \} \)
DB setup

• DB is memory resident => no I/O interference

• No dynamic and random parameters, no concurrency control among transactions
Workload choice

• Simple queries:
  – Single-table range selections (sequential, index)
  – Two-table equijoins
• Easy to setup and run
• Fully controllable parameters
• Isolates basic operations
• Enable iterative hypotheses !!!
• Building blocks for complex workloads?
Execution Time Breakdown (%)

- Stalls at least 50% of time
- Memory stalls are major bottleneck
Memory Stalls Breakdown (%)

- Role of L1 data cache and L2 instruction cache unimportant
- L2 data and L1 instruction stalls dominate
- Memory bottlenecks across DBMSs and queries vary
Effect of Record Size
10% Sequential Scan

- **L2D increase**: locality + page crossing (except D)
- **L1I increase**: page boundary crossing costs
Memory Bottlenecks

- **Memory is important**
  - Increasing memory-processor performance gap
  - Deeper memory hierarchies expected

- **Stalls due to L2 cache data misses**
  - Expensive fetches from main memory
  - L2 grows (8MB), but will be slower

- **Stalls due to L1 I-cache misses**
  - Buffer pool code is expensive
  - L1 I-cache not likely to grow as much as L2
Branch Mispredictions Are Expensive

- Rates are low, but contribution is significant
- A compiler task, but decisive for L1I performance
Mispredictions Vs. L1-I Misses

- More branch mispredictions incur more L1I misses
- Index code more complicated - needs optimization
Resource-related Stalls

- High $T_{DEP}$ for all systems: Low ILP opportunity
- A’s sequential scan: Memory unit load buffers?
Microbenchmarks vs. TPC

CPI Breakdown

- Sequential scan breakdown similar to TPC-D
- 2ary index and TPC-C: higher CPI, memory stalls (L2 D&I mostly)
Conclusions

• Execution time breakdown shows trends
• L1I and L2D are major memory bottlenecks
• We need to:
  – reduce page crossing costs
  – optimize instruction stream
  – optimize data placement in L2 cache
  – reduce stalls at all levels
• TPC may not be necessary to locate bottlenecks
Five years later – Becker et al 2004

- Same DBMSs, setup and workloads (memory resident) and same metrics
- Outcome: stalls still take lots of time
  - Seq scans: L1 stalls, branch mispredictions much lower
  - Index scans: no improvement
  - Joins: improvements, similar to seq scans
  - Bottleneck shift to L2D misses => must improve data placement
  - What works well on some hardware doesn’t on other
Five years later – Becker et al 2004

• C on a Quad P3 700MHz, 4G RAM, 16K L1, 2M L2
• B on a single P4 3GHz, 1G RAM, 8K L1D + 12KuOp trace cache, 512K L2, BTB 8x than P3

• P3 results:
  – Similar to 5 years ago: major bottlenecks are L1I and L2D

• P4 results:
  – Memory stalls almost entirely due to L1D and L2D stalls
  – L1D stalls higher - smaller cache and larger cache line
  – L1I stalls removed due to trace cache (esp. for seq. scan, but still some for index)

Hardware – awareness is important!
References

- Anastassia Ailamaki – VLDB’99 talk slides
Questions?