Understanding, modeling, and improving main-memory database performance

Manegold, S.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
List of Tables and Figures

Tables

3.1 Characteristic Parameters per Cache Level .................................. 49
3.2 Calibrated Performance Characteristics .................................. 66
4.1 Sample Data Access Patterns ............................................... 79
4.2 Hardware Characteristics .................................................. 99
5.1 Hardware Counters used for Execution Time Breakdown .......... 112

Figures

1.1 Query Processing Architecture ........................................... 12
2.1 Vertically Decomposed Storage in BATs ............................... 38
3.1 Trends in DRAM and CPU speed ....................................... 42
3.2 Modern CPU and Hierarchical Memory Architecture .............. 42
3.3 Sequential scan: Performance .......................................... 50
3.4 Sequential scan: Cache misses ......................................... 50
3.5 Seq. scan cache misses: Measured and Modeled .................... 52
3.6 Seq. scan performance: Experiment and Models ................... 52
3.7 CPU and memory access costs per tuple in a simple table scan ... 54
3.8 Vertical Decomposition in BATs ......................................... 56
3.9 Calibration Tool: Walking "backward" through the memory array .. 58
3.10 Calibration Tool: Cache sizes, line sizes, and miss latencies ... 60
3.11 Calibration Tool: Cache sizes, line sizes, and replace times ... 61
3.12 Calibration Tool: Cache associativities ............................... 63
3.13 Calibration Tool: TLB entries and TLB miss costs ................ 64
3.14 Three ways to add a buffer of integers, and costs per addition on the Origin2000 ................................................. 67
List of Tables and Figures

4.1 Single Sequential Traversal: $s.\text{trav}(R, u)$ ........................................... 76
4.2 Single Random Traversal: $r.\text{trav}(R, u)$ .................................................. 76
4.3 Interleaved Multi-Cursor Access: $\text{nest}(R, m, s.\text{trav}(R, u), \text{seq, bi})$ .... 77
4.4 Impact of gap-size on the Number of Cache Misses ........................................... 81
4.5 Impact of Alignment on the Number of Cache Misses ........................................ 82
4.6 Impact of $u$ and its Alignment on the Number of Cache Misses .......................... 84
4.7 Impact of length and width on the Number of Cache Misses ................................ 86
4.8 Measured and Predicted Cache Misses and Execution Time of Quick-Sort .............. 100
4.9 Measured and Predicted Cache Misses and Execution Time of Merge-Join .............. 101
4.10 Measured and Predicted Cache Misses and Execution Time of Hash-Join 102
4.11 Measured and Predicted Cache Misses and Execution Time of Partitioning and Partitioned Hash-Join ................................................................. 102

5.1 Straightforward cluster algorithm ................................................................. 109
5.2 2-pass/3-bit Radix Cluster ................................................................................. 109
5.3 Execution Time Breakdown of Radix-Cluster using one pass .............................. 113
5.4 Execution Time Breakdown of Radix-Cluster using optimal number of passes ...... 114
5.5 C language radix-cluster with annotated CPU optimizations ............................ 115
5.6 Execution Time Breakdown of optimized Radix-Cluster using one pass ............... 116
5.7 Execution Time Breakdown of optimized Radix-Cluster using optimal number of passes ........................................................................................................ 117
5.8 Measured and Modeled Events of Radix-Cluster (Origin2000) .......................... 119
5.9 Measured and Modeled Performance of Radix-Cluster ...................................... 120
5.10 Bit-distribution for 2-pass Radix-Cluster .......................................................... 121
5.11 Execution Time Breakdown of Partitioned Hash-Join ........................................ 122
5.12 C language hash-join with annotated CPU optimizations ............................... 123
5.13 Execution Time Breakdown of optimized Partitioned Hash-Join ....................... 124
5.14 Measured and Modeled Events of Partitioned Hash-Join (Origin2000) .............. 126
5.15 Measured and Modeled Time of Partitioned Hash-Join ....................................... 127
5.16 Overall Performance of Partitioned Hash-Join ............................................... 129
5.17 Measured and Modeled Overall Performance of Partitioned Hash-Join .............. 130
5.18 Execution Time Breakdown of Radix-Join ...................................................... 132
5.19 C language nested-loop with annotated CPU optimizations ........................... 133
5.20 Execution Time Breakdown of optimized Radix-Join ........................................ 133
5.21 Measured and Modeled Events and Performance of Radix-Join ....................... 135
5.22 Overall Performance of Radix-Join ..................................................................... 136
5.23 Partitioned Hash-Join vs. Radix-Join: non-optimized and optimized ................. 137