Full text available at: http://dx.doi.org/10.1561/1900000058

Main Memory Database Systems

Franz Faerber
SAP
franz.faerber@sap.com

Alfons Kemper
Technische Universität München
alfons.kemper@in.tum.de

Per-Åke Larson
Microsoft Research
palarson@acm.org

Justin Levandoski
Microsoft Research
justin.levandoski@microsoft.com

Thomas Neumann
Technische Universität München
thomas.neumann@in.tum.de

Andrew Pavlo
Carnegie Mellon University
pavlo@cs.cmu.edu

now
the essence of knowledge
Boston — Delft

Full text available at: http://dx.doi.org/10.1561/1900000058

Foundations and Trends® in Databases

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

F. Faerber, A. Kemper, P. Å Larson, J. Levandoski, T. Neumann, and A. Pavlo. *Main Memory Database Systems*. Foundations and Trends® in Databases, vol. 8, no. 1-2, pp. 1–130, 2016.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-324-9
© 2017 F. Faerber, A. Kemper, P. Å Larson, J. Levandoski, T. Neumann, and A. Pavlo

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The ‘services’ for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1 781 871 0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com

Full text available at: http://dx.doi.org/10.1561/1900000058

Foundations and Trends® in Databases

Volume 8, Issue 1-2, 2016

Editorial Board

Editor-in-Chief

Joseph M. Hellerstein
University of California, Berkeley
United States

Editors

Anastasia Ailamaki
EPFL

Mike Cafarella
University of Michigan

Michael Carey
UC Irvine

Surajit Chaudhuri
Microsoft Research

Minos Garofalakis
Yahoo! Research

Ihab Ilyas
University of Waterloo

Christopher Olston
Yahoo! Research

Jignesh Patel
University of Michigan

Chris Re
Stanford University

Gerhard Weikum
Max Planck Institute Saarbrücken

Full text available at: http://dx.doi.org/10.1561/1900000058

Editorial Scope

Topics

Foundations and Trends® in Databases covers a breadth of topics relating to the management of large volumes of data. The journal targets the full scope of issues in data management, from theoretical foundations, to languages and modeling, to algorithms, system architecture, and applications. The list of topics below illustrates some of the intended coverage, though it is by no means exhaustive:

- Data models and query languages
- Query processing and optimization
- Storage, access methods, and indexing
- Transaction management, concurrency control, and recovery
- Deductive databases
- Parallel and distributed database systems
- Database design and tuning
- Metadata management
- Object management
- Trigger processing and active databases
- Data mining and OLAP
- Approximate and interactive query processing
- Data warehousing
- Adaptive query processing
- Data stream management
- Search and query integration
- XML and semi-structured data
- Web services and middleware
- Data integration and exchange
- Private and secure data management
- Peer-to-peer, sensornet, and mobile data management
- Scientific and spatial data management
- Data brokering and publish/subscribe
- Data cleaning and information extraction
- Probabilistic data management

Information for Librarians

Foundations and Trends® in Databases, 2016, Volume 8, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.

Full text available at: http://dx.doi.org/10.1561/1900000058

Foundations and Trends® in Databases
Vol. 8, No. 1-2 (2016) 1–130
© 2017 F. Faerber, A. Kemper, P. Å Larson,
J. Levandoski, T. Neumann, and A. Pavlo
DOI: 10.1561/1900000058

Main Memory Database Systems

Franz Faerber
SAP
franz.faerber@sap.com

Alfons Kemper
Technische Universität München
alfons.kemper@in.tum.de

Per-Åke Larson
Microsoft Research
palarson@acm.org

Justin Levandoski
Microsoft Research
justin.levandoski@microsoft.com

Thomas Neumann
Technische Universität München
thomas.neumann@in.tum.de

Andrew Pavlo
Carnegie Mellon University
pavlo@cs.cmu.edu

Full text available at: http://dx.doi.org/10.1561/1900000058

Contents

1 Introduction 2

2 History and Trends 6
2.1 History 6
2.2 Trends 10

3 Issues and Architectural Choices 12
3.1 Data Organization and Layout 12
3.2 Indexing 18
3.3 Concurrency Control 23
3.4 Durability and Recovery 28
3.5 Query Processing and Compilation 31

4 Systems 35
4.1 SQL Server Hekaton 35
4.2 H-Store and VoltDB 47
4.3 HyPer 62
4.4 SAP HANA 86
4.5 Other Systems 101

5 Active Research and Future Directions 106
5.1 Cold Data Management 106

Full text available at: http://dx.doi.org/10.1561/1900000058

5.2 Transactional Memory 110
5.3 Non-Volatile RAM 112

References **115**

Full text available at: http://dx.doi.org/10.1561/1900000058

Abstract

This article provides an overview of recent developments in main-memory database systems. With growing memory sizes and memory prices dropping by a factor of 10 every 5 years, data having a "primary home" in memory is now a reality. Main-memory databases eschew many of the traditional architectural pillars of relational database systems that optimized for disk-resident data. The result of these memory-optimized designs are systems that feature several innovative approaches to fundamental issues (e.g., concurrency control, query processing) that achieve orders of magnitude performance improvements over traditional designs. Our survey covers five main issues and architectural choices that need to be made when building a high performance main-memory optimized database: data organization and storage, indexing, concurrency control, durability and recovery techniques, and query processing and compilation. We focus our survey on four commercial and research systems: H-Store/VoltDB, Hekaton, HyPer, and SAP HANA. These systems are diverse in their design choices and form a representative sample of the state of the art in main-memory database systems. We also cover other commercial and academic systems, along with current and future research trends.

F. Faerber, A. Kemper, P. Å Larson, J. Levandoski, T. Neumann, and A. Pavlo. *Main Memory Database Systems*. Foundations and Trends® in Databases, vol. 8, no. 1-2, pp. 1–130, 2016.

DOI: 10.1561/1900000058.

Full text available at: http://dx.doi.org/10.1561/1900000058

1

Introduction

Research and development of main-memory database systems started in the early eighties [37], with several commercial systems appearing in the nineties (e.g., TimesTen [146], P*Time [28], DataBlitz [16]). Many of these systems were used in targeted, performance-critical applications, mainly in telecommunications and finance. The price and capacity of memory during this time period limited applicability of many of these engines, thus main-memory systems did not - at the time - succeed as a general data processing solution.

Recently, two trends have made this field interesting again: memory prices and multi-core parallelism. For the last 30 years memory prices have dropped by a factor of 10 every 5 years. A server with 32 cores and 1 TB of memory now costs around $40K. Machines such as these make it feasible to fit most (if not all) of the world's OLTP workloads[1] comfortably into memory at a reasonable price. In addition, modern CPUs provide a staggering amount of raw parallelism. Vanilla CPUs contain at least 8 cores, and it is common for modern servers to contain two to four CPU sockets (16 to 32 cores). Core counts continue to rise, with Intel currently shipping a Xeon CPU with 18 cores [1].

Such parallelism coupled with the ability to (practically) store data completely in memory has brought about a recent flurry of research and development into main-memory databases. The result has been astounding. Prominent research systems such as H-Store [142] and

---

[1] The focus of this survey is primarily on main-memory OLTP databases.

Full text available at: http://dx.doi.org/10.1561/1900000058

HyPeR [72] reinvigorated research into main-memory and multi-core data processing techniques. Most major database vendors now have an in-memory database solution, such as SAP HANA [137], Oracle TimesTen [74], and Microsoft SQL Server Hekaton [38]. In addition, a number of startups such as VoltDB [143] and MemSQL [2] have carved out a niche in the database vendor landscape.

The result of this research and development is a new breed of database system with a radically different design when compared to a traditional disk-based relational system. These systems abandon many of the "textbook" design tenets in favor of new (or revisited) approaches to achieve high performance on modern hardware. For instance, the following examples provide an idea of how different these systems are:

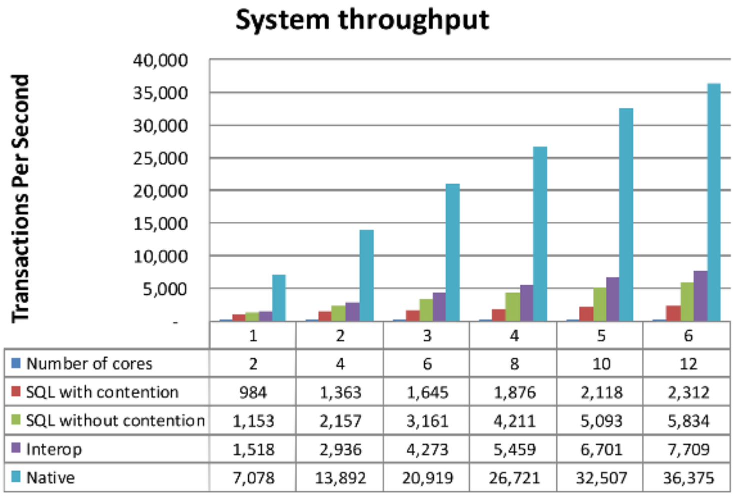
- **Data organization and indexing.** A pervasive trend in all systems is to avoid page-based indirection through a buffer pool and store only records in memory. Indexes usually store direct pointers to records. Several systems also implement novel indexing methods that optimize for CPU cache efficiency [85] as well as multi-core parallelism using latch-free designs [89].

- **Concurrency control.** Most systems avoid pessimistic lock-based concurrency control due to blocking and context switch overheads. Instead, some systems use a multi-version concurrency control variant [11, 77, 74, 111], while others use partitioned serial execution to achieve high performance [72, 142, 143].

- **Durability and recovery.** Aries-style redo/undo logging and recovery is rarely used. Instead, most systems opt for a form of redo-only logging (or command logging) coupled with periodic database snapshots to recover from a crash or restart [38, 74, 142].

- **Query processing and compilation.** To avoid the overhead of virtual function calls, degradation of branch prediction, and byte interpretation, several systems abandon the "get next" iterator processing model. Instead, queries are compiled into highly optimized machine code and run directly over in-memory records [38, 47, 109].

Full text available at: http://dx.doi.org/10.1561/1900000058



	1	2	3	4	5	6
🟦 Number of cores	2	4	6	8	10	12
🟥 SQL with contention	984	1,363	1,645	1,876	2,118	2,312
🟩 SQL without contention	1,153	2,157	3,161	4,211	5,093	5,834
🟪 Interop	1,518	2,936	4,273	5,459	6,701	7,709
🟦 Native	7,078	13,892	20,919	26,721	32,507	36,375

Figure 1.1: Microsoft Hekaton vs the traditional SQL Server engine taken from [38]. Latch contention plays a large part in limiting the scalability of the traditional engine.

The list above resembles a set topics once thought to be picked over and "closed" in the database research literature. However, over the past several years there has been significant innovation in these and other core areas with the reemergence of main-memory database systems, making it a vibrant and exciting technology space.

The innovations made in main-memory databases come with meaningful performance gains. Figure 1.1 provides an evaluation of the thread-level scalability of the Microsoft Hekaton engine compared to the traditional SQL Server engine running a typical customer workload[2] . Hekaton achieves a roughly 15.7X performance improvement at 12 cores, while the scalability of the traditional engine is limited due to the overheads inherent in a disk-based architecture running a memory-bound workload.

This article provides an overview of the research and development of modern main-memory database systems. We focus our sur-

---

[2] Graph taken from the original paper [38]

Full text available at: http://dx.doi.org/10.1561/1900000058

vey mainly on transaction processing engines through the lens of four main-memory systems: (1) *H-Store/VoltDB* [142, 143], a pioneering research system from academic research groups at MIT, Brown, Brandeis, and Yale that subsequently became a commercial database; (2) *Hekaton* [38], Microsoft SQL Server's main-memory OLTP engine; (3) *HyPeR* [72], a prominent research system from TU Munich that aims to support both high performance OLTP and OLAP workloads in the same engine; and (4) *SAP HANA* [137], the first main-memory optimized engine to ship from a major database vendor. As we will see, each of these systems are diverse in their design choices and form a representative sample of the state of the art in main-memory database systems. Our coverage focuses on five issues that influence the architecture and design of the system: (a) data storage and layout, (b) indexing and data structure design, (c) concurrency control, (d) durability and recovery, and (e) query processing and compilation. We also summarize the design of other modern commercial and academic main-memory systems.

The organization of this survey is as follows. Chapter 2 summarizes the past research and development of main-memory database systems prior to the "modern era" (starting around 2007). Chapter 3 summarize the issues used in our system survey. Chapter 4 provides case studies describing the design of our four representative systems. Chapter 5 concludes this survey by summarizing interesting current and future research trends in main-memory databases.

References

[1] Intel Xeon Processor E5-2699 v3 Specification Sheet. `http://intel.ly/1xx6aZz`.

[2] MemSQL. `http://www.memsql.com`.

[3] VoltDB. `http://www.voltdb.com`.

[4] Daniel J. Abadi, Samuel Madden, and Nabil Hachem. Column-Stores vs. Row-Stores: How Different are they Really? In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 967–980, 2008.

[5] Atul Adya, Robert Gruber, Barbara Liskov, and Umesh Maheshwari. Efficient Optimistic Concurrency Control using Loosely Synchronized Clocks. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 23–34, 1995.

[6] Rakesh Agrawal, Michael J. Carey, and Miron Livny. Concurrency control performance modeling: Alternatives and implications. *ACM Trans. Database Syst.*, 12(4):609–654, 1987.

[7] Rakesh Agrawal and David J. DeWitt. Integrated Concurrency Control and Recovery Mechanisms: Design and Performance Evaluation. *ACM Transactions on Database Systems, TODS*, 10(4):529–564, 1985.

[8] Anastassia Ailamaki, David J. DeWitt, Mark D. Hill, and David A. Wood. Dbmss on a modern processor: Where does time go? VLDB, pages 266–277, 1999.

Full text available at: http://dx.doi.org/10.1561/1900000058

[9] M.-C. Albutiu, Alfons Kemper, and Thomas Neumann. Massively Parallel Sort-Merge Joins in Main Memory Multi-Core Database Systems. *Proceedings of the VLDB Endowment, PVLDB*, 5(10):1064–1075, 2012.

[10] Karolina Alexiou, Donald Kossmann, and Per-Åke Larson. Adaptive Range Filters for Cold Data: Avoiding Trips to Siberia. *Proceedings of the VLDB Endowment, PVLDB*, 6(14):1714–1725, 2013.

[11] Altibase. *Altibase HDB Administrator's Manual*, 5 2015.

[12] Peter M. G. Apers, Martin L. Kersten, and Hans Oerlemans. PRISMA Database Machine: A Distributed, Main-Memory Approach. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, pages 590–593, 1988.

[13] Peter M. G. Apers, Carel A. van den Berg, Jan Flokstra, Paul W. P. J. Grefen, Martin L. Kersten, and Annita N. Wilschut. PRISMA/DB: A parallel main memory relational DBMS. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 4(6):541–554, 1992.

[14] Joy Arulraj, Andrew Pavlo, and Subramanya Dulloor. Let's Talk About Storage & Recovery Methods for Non-Volatile Memory Database Systems. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 707–722, 2015.

[15] Cagri Balkesen, Jens Teubner, Gustavo Alonso, and M. Tamer Özsu. Main-Memory Hash Joins on Multi-Core CPUs: Tuning to the Underlying Hardware. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 362–373, 2013.

[16] Jerry Baulier, Philip Bohannon, S. Gogate, C. Gupta, S. Haldar, S. Joshi, A. Khivesera, Henry F. Korth, Peter McIlroy, J. Miller, P. P. S. Narayan, M. Nemeth, Rajeev Rastogi, S. Seshadri, Abraham Silberschatz, S. Sudarshan, M. Wilder, and C. Wei. DataBlitz Storage Manager: Main Memory Database Performance for Critical Applications. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 519–520, 1999.

[17] Philip A. Bernstein and Nathan Goodman. Timestamp-Based Algorithms for Concurrency Control in Distributed Database Systems. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 285–300, 1980.

[18] Philip A. Bernstein and Nathan Goodman. Concurrency Control in Distributed Database Systems. *ACM Comput. Surv.*, 13(2):185–221, 1981.

Full text available at: http://dx.doi.org/10.1561/1900000058

[19] Philip A. Bernstein and Nathan Goodman. A sophisticate's introduction to distributed concurrency control (invited paper). In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 62–76, 1982.

[20] Philip A. Bernstein, Vassos Hadzilacos, and Nathan Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[21] Dina Bitton, Maria Hanrahan, and Carolyn Turbyfill. Performance of Complex Queries in Main Memory Database Systems. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 72–81, 1987.

[22] Spyros Blanas, Yinan Li, and Jignesh M. Patel. Design and Evaluation of Main Memory Hash Join Algorithms for Multi-core CPUs. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 37–48, 2011.

[23] Philip Bohannon, Daniel F. Lieuwen, Rajeev Rastogi, Abraham Silberschatz, S. Seshadri, and S. Sudarshan. The Architecture of the Dalí Main-Memory Storage Manager. *Multimedia Tools Appl.*, 4(2):115–151, 1997.

[24] Michael J. Carey and Miron Livny. Distributed Concurrency Control Performance: A Study of Algorithms, Distribution, and Replication. Proceedings of the International Conference on Very Large Data Bases, VLDB, pages 13–25, 1988.

[25] Michael J. Carey and Michael Stonebraker. The Performance of Concurrency Control Algorithms for Database Management Systems. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 107–118, 1984.

[26] Rick Cattell. Scalable SQL and NoSQL Data Stores. *SIGMOD Record*, 39:12–27, 2011.

[27] Sang Kyun Cha, Sangyong Hwang, Kihong Kim, and Keunjoo Kwon. Cache-Conscious Concurrency Control of Main-Memory Indexes on Shared-Memory Multiprocessor Systems. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 181–190, 2001.

[28] Sang Kyun Cha and Changbin Song. P*TIME: Highly Scalable OLTP DBMS for Managing Update-Intensive Stream Workload. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 1033–1044, 2004.

Full text available at: http://dx.doi.org/10.1561/1900000058

[29] Andreas Chatzistergiou, Marcelo Cintra, and Stratis D. Viglas. REWIND: Recovery Write-Ahead System for In-Memory Non-Volatile Data-Structures. *Proceedings of the VLDB Endowment, PVLDB*, 8(5):497–508, 2015.

[30] Shimin Chen, Phillip B. Gibbons, and Todd C. Mowry. Improving index performance through prefetching. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 235–246, 2001.

[31] Joel Coburn, Trevor Bunker, Meir Schwarz, Rajesh Gupta, and Steven Swanson. From ARIES to MARS: Transaction Support for Next-Generation, Solid-State Drives. In *Proceedings of the ACM Symposium on Operating Systems Principles, SOSP*, pages 197–212, 2013.

[32] James Cowling and Barbara Liskov. Granola: Low-Overhead Distributed Transaction Coordination. In *Proceedings of the USENIX Annual Technical Conference, ATC*, pages 21–34, June 2012.

[33] Carlo Curino, Evan Jones, Yang Zhang, and Sam Madden. Schism: a Workload-Driven Approach to Database Replication and Partitioning. *Proceedings of the International Conference on Very Large Data Bases, VLDB*, 3:48–57, 2010.

[34] Justin DeBrabant, Andrew Pavlo, Stephen Tu, Michael Stonebraker, and Stan Zdonik. Anti-caching: A New Approach to Database Management System Architecture. *Proceedings of the VLDB Endowment, PVLDB*, 6(14):1942–1953, 2013.

[35] Kalen Delaney, Paul Randal, Kimberley Tripp, Conor Cunningham, Adam Machanic, and Ben Navarez. *SQL Server 2008 Internals*. Microsoft, 2009.

[36] David J. DeWitt, Robert H. Gerber, Goetz Graefe, Michael L. Heytens, Krishna B. Kumar, and M. Muralikrishna. GAMMA - A High Performance Dataflow Database Machine. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 228–237, 1986.

[37] David J. DeWitt, Randy H. Katz, Frank Olken, Leonard D. Shapiro, Michael Stonebraker, and David A. Wood. Implementation Techniques for Main Memory Database Systems. *SIGMOD Record*, 14(2):1–8, 1984.

[38] Cristian Diaconu, Craig Freedman, Erik Ismert, Per-Åke Larson, Pravin Mittal, Ryan Stonecipher, Nitin Verma, and Mike Zwilling. Hekaton: SQL Server's Memory-Optimized OLTP Engine. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 1243–1254, 2013.

Full text available at: http://dx.doi.org/10.1561/1900000058

[39] Aleksandar Dragojević, Dushyanth Narayanan, Miguel Castro, and Orion Hodson. FaRM: Fast Remote Memory. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation, NSDI*, pages 401–414, 2014.

[40] Alexander Driskill-Smith. Latest Advances and Future Prospects of STT-RAM. In *Non-Volatile Memories Workshop*, 2010.

[41] Margaret H. Eich. A Classification and Comparison of Main Memory Database Recovery Techniques. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 332–339, 1987.

[42] Ahmed Eldawy, Justin Levandoski, and Per-Åke Larson. Trekking Through Siberia: Managing Cold Data in a Memory-Optimized Database. *Proceedings of the VLDB Endowment, PVLDB*, 7(11):931–942, 2014.

[43] Jose M. Faleiro and Daniel J. Abadi. Rethinking serializable multiversion concurrency control. *Proceedings of the VLDB Endowment, PVLDB*, 8(11):1190–1201, 2015.

[44] Jose M. Faleiro and Daniel J. Abadi. Latch-free Synchronization in Database Systems: Silver Bullet or Fool's Gold? In *Proceedings of the International Conference on Innovative Data Systems Research, CIDR*, 2017.

[45] Ru Fang, Hui-I Hsiao, Bin He, C. Mohan, and Yun Wang. High Performance Database Logging using Storage Class Memory. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 1221–1231, 2011.

[46] Peter A. Franaszek and John T. Robinson. Limitations of concurrency in transaction processing. *ACM Transactions on Database Systems, TODS*, 10(1):1–28, 1985.

[47] Craig Freedman, Erik Ismert, and Per-Åke Larson. Compilation in the Microsoft SQL Server Hekaton Engine. *IEEE Data Engineering Bulletin*, 37(1):22–30, 2014.

[48] Florian Funke, Alfons Kemper, and Thomas Neumann. Compacting Transactional Data in Hybrid OLTP & OLAP Databases. *Proceedings of the VLDB Endowment, PVLDB*, 5(11):1424–1435, 2012.

[49] Hector Garcia-Molina and Kenneth Salem. Main Memory Database Systems: An Overview. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 4(6):509–516, 1992.

[50] Dieter Gawlick and David Kinkade. Varieties of Concurrency Control in IMS/VS Fast Path. *IEEE Data Engineering Bulletin*, 8(2):3–10, 1985.

Full text available at: http://dx.doi.org/10.1561/1900000058

[51] Shahram Ghandeharizadeh, David J. DeWitt, and Waheed Qureshi. A Performance Analysis of Alternative Multi-Attribute Declustering Strategies. *Proceedings of the ACM Conference on Management of Data, SIGMOD*, 21(2):29–38, 1992.

[52] Vibby Gottemukkala and Tobin J. Lehman. Locking and Latching in a Memory-Resident Database System. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 533–544, 1992.

[53] Goetz Graefe. Query Evaluation Techniques for Large Databases. *ACM Computing Surveys*, 25(2):73–170, 1993.

[54] Goetz Graefe and William J. McKenna. The Volcano Optimizer Generator: Extensibility and Efficient Search. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 209–218, 1993.

[55] Jim Gray, Raymond A. Lorie, Gianfranco R. Putzolu, and Irving L. Traiger. Granularity of Locks in a Large Shared Data Base. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 428–451, 1975.

[56] Jim Gray and Gianfranco R. Putzolu. The 5 Minute Rule for Trading Memory for Disk Accesses and The 10 Byte Rule for Trading Memory for CPU Time. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 395–398, 1987.

[57] Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.

[58] Le Gruenwald and Margaret H. Eich. MMDB Reload Algorithms. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 397–405, 1991.

[59] Theo Haerder and Andreas Reuter. Principles of Transaction-Oriented Database Recovery. *ACM Computing Surveys*, 15(4):287–317, 1983.

[60] Stavros Harizopoulos, Daniel J. Abadi, Samuel Madden, and Michael Stonebraker. OLTP through the looking glass, and what we found there. In *SIGMOD*, pages 981–992, 2008.

[61] Pat Helland, Harald Sammer, Jim Lyon, Richard Carr, Phil Garrett, and Andreas Reuter. Group Commit Timers and High Volume Transaction Systems. In *Proceedings of the International Workshop on High Performance Transaction Systems, HPTS*, pages 301–329, 1989.

[62] Joseph M. Hellerstein and Michael Stonebraker. Readings in Database Systems. chapter Transaction Management, pages 238–243. 4th edition, 1998.

Full text available at: http://dx.doi.org/10.1561/1900000058

[63] Joseph M. Hellerstein, Michael Stonebraker, and James R. Hamilton. Architecture of a Database System. *Foundations and Trends in Databases*, 1(2):141–259, 2007.

[64] Maurice Herlihy and J. Eliot B. Moss. Transactional Memory: Architectural Support for Lock-Free Data Structures. In *Proceedings of the International Symposium on Computer Architecture, ISCA*, pages 289–300, 1993.

[65] M. Heytens, S. Listgarten, M-A. Neimat, and K. Wilkinson. Smallbase: A main-memory dbms for high-performance applications. Technical report, Hewlett-Packard Laboratories, 1995.

[66] Svein-Olaf Hvasshovd, Øystein Torbjørnsen, Svein Erik Bratsberg, and Per Holager. The ClustRa Telecom Database: High Availability, High Throughput, and Real-Time Response. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 469–477, 1995.

[67] H. V. Jagadish, Daniel F. Lieuwen, Rajeev Rastogi, Abraham Silberschatz, and S. Sudarshan. Dalí: A High Performance Main Memory Storage Manager. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 48–59, 1994.

[68] Ryan Johnson, Ippokratis Pandis, Radu Stoica, Manos Athanassoulis, and Anastasia Ailamaki. Aether: A Scalable Approach to Logging. *Proceedings of the VLDB Endowment, PVLDB*, 3(1-2):681–692, 2010.

[69] J. R. Jordan, J. Banerjee, and R. B. Batman. Precision Locks. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 143–147, 1981.

[70] Robert Kallman, Hideaki Kimura, Jonathan Natkins, Andrew Pavlo, Alexander Rasin, Stanley Zdonik, Evan P. C. Jones, Samuel Madden, Michael Stonebraker, Yang Zhang, John Hugg, and Daniel J. Abadi. H-Store: A High-Performance, Distributed Main Memory Transaction Processing System. *Proceedings of the VLDB Endowment, PVLDB*, 1(2):1496–1499, 2008.

[71] Tomas Karnagel, Roman Dementiev, Ravi Rajwar, Konrad Lai, Thomas Legler, Benjamin Schlegel, and Wolfgang Lehne. Improving In-Memory Database Index Performance with Intel Transactional Synchronization Extensions. In *Proceedings of the IEEE Symposium on High Performance Computer Architecture, HPCA*, pages 476–487, 2014.

Full text available at: http://dx.doi.org/10.1561/1900000058

[72] Alfons Kemper and Thomas Neumann. HyPer: A Hybrid OLTP&OLAP Main Memory Database System Based on Virtual Memory Snapshots. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 195–206, 2011.

[73] Konstantinos Krikellas, Stratis Viglas, and Marcelo Cintra. Generating Code for Holistic Query Evaluation. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 613–624, 2010.

[74] Tirthankar Lahiri, Marie-Anne Neimat, and Steve Folkman. Oracle TimesTen: An In-Memory Database for Enterprise Applications. *IEEE Data Engineering Bulletin*, 36(2):6–13, 2013.

[75] Harald Land, Tobias Mühlbauer, Florian Funke, Peter Boncz, Thomas Neumann, and Alfons Kemper. Data Blocks: Hybrid OLTP and OLAP on Compressed Storage using both Vectorization and Compilation. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 311–326, 2016.

[76] Harald Lang, Viktor Leis, Martina-Cezara Albutiu, Thomas Neumann, and Alfons Kemper. Massively Parallel NUMA-aware Hash Joins. In *Proceedings of the International Workshop on In-Memory Data Management, IMDM*, 2013.

[77] Per-Åke Larson, Spyros Blanas, Cristian Diaconu, Craig Freedman, Jignesh M. Patel, and Mike Zwilling. High-Performance Concurrency Control Mechanisms for Main-Memory Databases. *Proceedings of the VLDB Endowment, PVLDB*, 5(4):298–309, 2011.

[78] Juchang Lee, Michael Muehle, Norman May, Franz Faerber, Vishal Sikka, Hasso Plattner, Jens Krüger, and Martin Grund. High-Performance Transaction Processing in SAP HANA. *IEEE Data Engineering Bulletin*, 36(2):28–33, 2013.

[79] Tobin J. Lehman and Michael J. Carey. A Study of Index Structures for Main Memory Database Management Systems. In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 294–303, 1986.

[80] Tobin J. Lehman and Michael J. Carey. Query Processing in Main Memory Database Management Systems. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 239–250, 1986.

[81] Tobin J. Lehman and Michael J. Carey. A Recovery Algorithm for a High-Performance Memory-Resident Database System. Proceedings of the ACM Conference on Management of Data, SIGMOD, pages 104–117, 1987.

[82] Tobin J. Lehman, Eugene J. Shekita, and Luis-Felipe Cabrera. An Evaluation of Starburst's Memory Resident Storage Component. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 4(6):555–566, 1992.

[83] Viktor Leis, Peter A. Boncz, Alfons Kemper, and Thomas Neumann. Morsel-Driven Parallelism: A NUMA-Aware Query Evaluation Framework for the Many-core Age. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 743–754, 2014.

[84] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter A. Boncz, Alfons Kemper, and Thomas Neumann. How Good Are Query Optimizers, Really? *Proceedings of the VLDB Endowment, PVLDB*, 9(3):204–215, 2015.

[85] Viktor Leis, Alfons Kemper, and Thomas Neumann. The Adaptive Radix Tree: ARTful Indexing for Main-Memory Databases. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[86] Viktor Leis, Alfons Kemper, and Thomas Neumann. Exploiting Hardware Transactional Memory in Main-Memory Databases. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 580–591, 2014.

[87] Viktor Leis, Kan Kundhikanjana, Alfons Kemper, and Thomas Neumann. Efficient Processing of Window Functions in Analytical SQL Queries. *Proceedings of the VLDB Endowment, PVLDB*, 8(10):1058–1069, 2015.

[88] Justin Levandoski, Per-Åke Larson, and Radu Stoica. Identifying Hot and Cold Data in Main-Memory Databases. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 26–37, 2013.

[89] Justin Levandoski, David B. Lomet, and Sudipta Sengupta. The Bw-Tree: A B-tree for New Hardware Platforms. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 302–313, 2013.

[90] Kai Li and Jeffrey F. Naughton. Multiprocessor Main Memory Transaction Processing. *Proceedings of the International Symposium on Databases in Parallel and Distributed Systems, DPDS*, pages 177–187, 1988.

[91] Antti-Pekka Liedes and Antoni Wolski. SIREN: A Memory-Conserving, Snapshot-Consistent Checkpoint Algorithm for in-Memory Databases. In *Proceedings of the International Conference on Data Engineering, ICDE*, page 99, 2006.

Full text available at: http://dx.doi.org/10.1561/1900000058

[92] Hyeontaek Lim, Dongsu Han, David G. Andersen, and Michael Kaminsky. MICA: A Holistic Approach to Fast In-Memory Key-Value Storage. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation, NSDI*, pages 429–444, 2014.

[93] Jan Lindström, Vilho Raatikka, Jarmo Ruuth, Petri Soini, and Katriina Vakkila. IBM solidDB: In-Memory Database Optimized for Extreme Speed and Availability. *IEEE Data Engineering Bulletin*, 36(2):14–20, 2013.

[94] David B. Lomet. Process Structuring, Synchronization, and Recovery Using Atomic Actions. In *Proceedings of the ACM Conference on Language Design for Reliable Software*, pages 128–137, 1977.

[95] Darko Makreshanski, Justin Levandoski, and Ryan Stutsman. To Lock, Swap, or Elide: On the Interplay of Hardware Transactional Memory and Lock-Free Indexing. *Proceedings of the VLDB Endowment, PVLDB*, 8(11):1298–1309, 2015.

[96] Nirmesh Malviya, Ariel Weisberg, Samuel Madden, and Michael Stonebraker. Rethinking main memory OLTP recovery. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 604–615, 2014.

[97] Yandong Mao, Eddie Kohler, and Robert Tappan Morris. Cache Craftiness for Fast Multicore Key-Value Storage. In *Proceedings of the European Conference on Computer Systems, EuroSys*, pages 183–196, 2012.

[98] Norman May, Wolfgang Lehner, Shahul Hameed P., Nitesh Maheshwari, Carsten Müller, Sudipto Chowdhuri, and Anil K. Goel. SAP HANA - From Relational OLAP Database to Big Data Infrastructure. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, pages 581–592, 2015.

[99] Memcached. `http://memcached.org/`.

[100] MemSQL Durability and Recovery. `http://bit.ly/1nqM2qK`.

[101] Guido Moerkotte, David DeHaan, Norman May, Anisoara Nica, and Alexander Böhm. Exploiting Ordered Dictionaries to Efficiently Construct Histograms with Q-error Guarantees in SAP HANA. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 361–372, 2014.

[102] C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, and Peter Schwarz. ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks using Write-Ahead Logging. *ACM Transactions on Database Systems, TODS*, 17(1):94–162, 1992.

Full text available at: http://dx.doi.org/10.1561/1900000058

[103] C. Mohan, B. Lindsay, and R. Obermarck. Transaction Management in the R* Distributed Database Management System. *ACM Transactions on Database Systems, TODS*, 11(4):378–396, 1986.

[104] Henrik Mühe, Alfons Kemper, and Thomas Neumann. How to Efficiently Snapshot Transactional Data: Hardware or Software Controlled? In *Proceedings of the International Workshop on Data Management on New Hardware, DaMoN*, pages 17–26, 2011.

[105] Henrik Mühe, Alfons Kemper, and Thomas Neumann. The Mainframe Strikes Back: Elastic Multi-Tenancy using Main Memory Database Systems on a Many-core Server. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, pages 578–581, 2012.

[106] Tobias Mühlbauer, Wolf Rödiger, Angelika Reiser, Alfons Kemper, and Thomas Neumann. ScyPer: Elastic OLAP Throughput on Transactional Data. In *Proceedings of the ACM SIGMOD Workshop on Data Analytics in the Cloud, DanaC*, 2013.

[107] Ingo Müller, Cornelius Ratsch, and Franz Färber. Adaptive String Dictionary Compression in In-Memory Column-Store Database Systems. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, pages 283–294, 2014.

[108] Ingo Müller, Peter Sanders, Arnaud Lacurie, Wolfgang Lehner, and Franz Färber. Cache-Efficient Aggregation: Hashing Is Sorting. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 1123–1136, 2015.

[109] Thomas Neumann. Efficiently Compiling Efficient Query Plans for Modern Hardware. *Proceedings of the VLDB Endowment, PVLDB*, 4(9):539–550, 2011.

[110] Thomas Neumann and Alfons Kemper. Unnesting Arbitrary Queries. In *Datenbanksysteme für Business, Technologie und Web, BTW*, pages 383–402, 2015.

[111] Thomas Neumann, Tobias Mühlbauer, and Alfons Kemper. Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 677–689, 2015.

[112] Christos Nikolaou, Manolis Marazakis, and G. Georgiannakis. Transaction Routing for Distributed OLTP Systems: Survey and Recent Results. *Inf. Sci.*, 97:45–82, 1997.

Full text available at: http://dx.doi.org/10.1561/1900000058

[113] Chris Nyberg, Tom Barclay, Zarka Cvetanovic, Jim Gray, and David B. Lomet. Alphasort: A RISC Machine Sort. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 233–242, 1994.

[114] Diego Ongaro, Stephen M. Rumble, Ryan Stutsman, John K. Ousterhout, and Mendel Rosenblum. Fast Crash Recovery in RAMCloud. In *Proceedings of the ACM Symposium on Operating Systems Principles, SOSP*, pages 29–41, 2011.

[115] Ismail Oukid, Daniel Booss, Wolfgang Lehner, Peter Bumbulis, and Thomas Willhalm. SOFORT: a Hybrid SCM-DRAM Storage Engine for Fast Data Recovery. In *Proceedings of the International Workshop on Data Management on New Hardware, DaMoN*, pages 1–7, 2014.

[116] John Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Guru Parulkar, Mendel Rosenblum, Stephen M. Rumble, Eric Stratmann, and Ryan Stutsman. The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM. *Operating Systems Review*, 43(4):92–105, 2009.

[117] John K. Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David Mazières, Subhasish Mitra, Aravind Narayanan, Diego Ongaro, Guru M. Parulkar, Mendel Rosenblum, Stephen M. Rumble, Eric Stratmann, and Ryan Stutsman. The Case for RAMCloud. *Communications of the ACM*, 54(7):121–130, 2011.

[118] Sriram Padmanabhan. *Data Placement in Shared-Nothing Parallel Database Systems*. PhD thesis, University of Michigan, 1992.

[119] Ippokratis Pandis, Pinar Tözün, Ryan Johnson, and Anastasia Ailamaki. PLP: Page Latch-free Shared-everything OLTP. *Proc. VLDB Endow.*, 4(10):610–621, July 2011.

[120] Andrew Pavlo, Carlo Curino, and Stanley B. Zdonik. Skew-Aware Automatic Data Partitioning in Shared-Nothing, Parallel OLTP Systems. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, 2012.

[121] Steven Pelley, Thomas F. Wenisch, Brian T. Gold, and Bill Bridge. Storage Management in the NVRAM Era. *Proceedings of the VLDB Endowment, PVLDB*, 7(2):121–132, 2013.

[122] Slawomir Pilarski and Tiko Kameda. Checkpointing for Distributed Databases: Starting from the Basics. *IEEE Transactions on Parallel Distributed Systems, TPDS*, 3(5):602–610, 1992.

Full text available at: http://dx.doi.org/10.1561/1900000058

[123] Holger Pirk, Florian Funke, Martin Grund, Thomas Neumann, Ulf Leser, Stefan Manegold, Alfons Kemper, and Martin L. Kersten. CPU and Cache Efficient Management of Memory-Resident Databases. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 14–25, 2013.

[124] Hasso Plattner. A Common Database Approach for OLTP and OLAP using an In-Memory Column Database. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 1–2, 2009.

[125] Iraklis Psaroudakis, Florian Wolf, Norman May, Thomas Neumann, Alexander Böhm, Anastasia Ailamaki, and Kai-Uwe Sattler. Scaling Up Mixed Workloads: A Battle of Data Freshness, Flexibility, and Scheduling. In *Proceedings of the Technology Conference on Performance Evaluation and Benchmarking, TPCTC*, pages 97–112, 2014.

[126] Erhard Rahm. A Framework for Workload Allocation in Distributed Transaction Processing Systems. *Journal on Systems Software*, 18:171–190, May 1992.

[127] Jun Rao and Kenneth A. Ross. Making b^+-trees cache conscious in main memory. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 475–486, 2000.

[128] Simone Raoux, Geoffrey W. Burr, Matthew J. Breitwisch, Charles T. Rettner, Yi-Chou Chen, Robert M. Shelby, Martin Salinga, Daniel Krebs, S-H Chen, Hsiang-Lan Lung, and Charles Lam. Phase-Change Random Access Memory: A Scalable Technology. *IBM Journal of Research and Development*, 52(4.5):465–479, 2008.

[129] Kun Ren, Thaddeus Diamond, Daniel J. Abadi, and Alexander Thomson. Low-Overhead Asynchronous Checkpointing in Main-Memory Database Systems. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 1539–1551, 2016.

[130] Kun Ren, Alexander Thomson, and Daniel J. Abadi. Lightweight Locking for Main Memory Database Systems. *PVLDB*, 6(2):145–156, 2012.

[131] Wolf Rödiger, Tobias Mühlbauer, Alfons Kemper, and Thomas Neumann. High-Speed Query Processing over High-Speed Networks. *Proceedings of the VLDB Endowment, PVLDB*, 9(4):228–239, 2015.

[132] Wolf Rödiger, Tobias Mühlbauer, Philipp Unterbrunner, Angelika Reiser, Alfons Kemper, and Thomas Neumann. Locality-Sensitive Operators for Parallel Main-memory Database Clusters. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 592–603, 2014.

Full text available at: http://dx.doi.org/10.1561/1900000058

[133] Stephen M. Rumble, Ankita Kejriwal, and John K. Ousterhout. Log-Structured Memory for DRAM-based Storage. In *Proceedings of the USENIX conference on File and Storage Technologies, FAST*, pages 1–16, 2014.

[134] Kenneth Salem and Hector Garcia-Molina. Checkpointing Memory-Resident Databases. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 452–462, 1989.

[135] Kenneth Salem and Hector Garcia-Molina. System M: A Transaction Processing Testbed for Memory Resident Data. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 2(1):161–172, 1990.

[136] Rajkumar Sen, Jack Chen, and Nika Jimsheleishvilli. Query Optimization Time: The New Bottleneck in Real-time Analytics. In *Proceedings of the International Workshop on In-Memory Data Management, IMDM*, pages 1–6, 2015.

[137] Vishal Sikka, Franz Färber, Anil K. Goel, and Wolfgang Lehner. SAP HANA: The Evolution from a Modern Main-Memory Data Platform to an Enterprise Application Platform. *Proceedings of the VLDB Endowment, PVLDB*, 6(11):1184–1185, 2013.

[138] Vishal Sikka, Franz Färber, Wolfgang Lehner, Sang Kyun Cha, Thomas Peh, and Christof Bornhövd. Efficient Transaction Processing in SAP HANA Database: The End of a Column Store Myth. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 731–742, 2012.

[139] IBM Knowledge Center: Pessimistic vs. Optimistic Concurrency Control. `http://ibm.co/1PCPlI3`.

[140] Michael Stonebraker. The Case for Shared Nothing. *IEEE Data Engineering Bulletin*, 9:4–9, 1986.

[141] Michael Stonebraker and Ugur Çetintemel. "One Size Fits All": An Idea Whose Time Has Come and Gone. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 2–11, 2005.

[142] Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos, Nabil Hachem, and Pat Helland. The End of an Architectural Era: (It's Time for a Complete Rewrite). In *Proceedings of the International Conference on Very Large Data Bases, VLDB*, pages 1150–1160, 2007.

[143] Michael Stonebraker and Ariel Weisberg. The VoltDB Main Memory DBMS. *IEEE Data Engineering Bulletin*, 36(2):21–27, 2013.

Full text available at: http://dx.doi.org/10.1561/1900000058

[144] Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart, and R. Stanley Williams. The Missing Memristor Found. *Nature*, 453(7191):80–83, 2008.

[145] Y. C. Tay, Nathan Goodman, and Rajan Suri. Locking performance in centralized databases. *ACM Transactions on Database Systems, TODS*, 10(4):415–462, 1985.

[146] Times-Ten Team. In-Memory Data Management for Consumer Transactions The Times-Ten Approach. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 528–529, 1999.

[147] Times-Ten Team. In-Memory Data Management in the Application Tier. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 637–641, 2000.

[148] Times-Ten Team. Mid-tier Caching: The TimesTen Approach. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 588–593, 2002.

[149] Robert H. Thomas. A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases. *ACM Transactions on Database Systems, TODS*, 4(2):180–209, June 1979.

[150] Alexander Thomson, Thaddeus Diamond, Shu-Chun Weng, Kun Ren, Philip Shao, and Daniel J. Abadi. Calvin: Fast Distributed Transactions for Partitioned Database Systems. In *Proceedings of the ACM Conference on Management of Data, SIGMOD*, pages 1–12, 2012.

[151] Stephen Tu, Wenting Zheng, Eddie Kohler, Barbara Liskov, and Samuel Madden. Speedy Transactions in Multicore In-Memory Databases. In *Proceedings of the ACM Symposium on Operating Systems Principles, SOSP*, pages 18–32, 2013.

[152] Tianzheng Wang and Ryan Johnson. Scalable Logging through Emerging Non-Volatile Memory. *Proceedings of the VLDB Endowment, PVLDB*, 7(10):865–876, 2014.

[153] Zhaoguo Wang, Hao Qian, Jinyang Li, and Haibo Chen. Using Restricted Transactional Memory to Build a Scalable In-Memory Database. In *Proceedings of the European Conference on Computer Systems, EuroSys*, pages 1–26, 2014.

[154] Xingda Wei, Jiaxin Shi, Yanzhe Chen, Rong Chen, and Haibo Chen. Fast In-Memory Transaction Processing using RDMA and HTM. In *Proceedings of the ACM Symposium on Operating Systems Principles, SOSP*, pages 87–104, 2015.

Full text available at: http://dx.doi.org/10.1561/1900000058

[155] Gerhard Weikum and Gottfried Vossen. *Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery*. Morgan Kaufmann, 2002.

[156] Kyu-Young Whang and Ravi Krishnamurthy. Query Optimization in a Memory-resident Domain Relational Calculus Database System. *ACM Transactions on Database Systems, TODS*, 15(1):67–95, March 1990.

[157] Authur Whitney, Dennis Shasha, and Stevan Apter. High Volume Transaction Processing without Concurrency Control, Two Phase Commit, SQL or C++. In *Proceedings of the International Workshop on High Performance Transaction Systems, HPTS*, 1997.

[158] Thomas Willhalm, Ismail Oukid, Ingo Müller, and Franz Faerber. Vectorizing Database Column Scans with Complex Predicates. In *Proceedings of the International Workshop on Accelerating Data Management Systems Using Modern Processor and Storage Architectures, ADMS*, pages 1–12, 2013.

[159] Wenting Zheng, Stephen Tu, Eddie Kohler, and Barbara Liskov. Fast Databases with Fast Durability and Recovery Through Multicore Parallelism. In *Proceedings of the USENIX Symposium on Operation System Design and Implementation, OSDI*, pages 465–477, 2014.

[160] Daniel C. Zilio. *Physical Database Design Decision Algorithms and Concurrent Reorganization for Parallel Database Systems*. PhD thesis, University of Toronto, 1998.