



Contents

Acknowledgements	i
Abstract (English/Deutsch)	v
Table of Contents	ix
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Data Management	1
1.2 Evolution of Hardware	2
1.3 OLTP on Modern Hardware	3
1.4 Scaling Up on Multicores	4
1.5 Utilizing Resources within a Core	5
1.6 Thesis Statement and Contributions	6
1.7 Roadmap	7
2 Background	9
2.1 Transaction Processing	9
2.2 Micro-architecture of OLTP's Playground	11
2.3 Exploiting Modern Hardware while Running OLTP	13
2.3.1 Scaling Up OLTP	13
2.3.2 Minimizing Memory Stalls	15
2.4 Evolution of TPC's OLTP benchmarks	17
2.4.1 The obsolete TPC-A and TPC-B	17
2.4.2 The ubiquitous TPC-C	18
2.4.3 The unexplored TPC-E	20
2.4.4 The evolution summary	22
2.5 The TATP benchmark	23
2.6 Shore-MT and Shore-Kits: Benchmarks on Top of Shore-MT	24

I Scalable and Dynamically Balanced Shared-Everything OLTP with Physiological Partitioning **25**

3 Latch-free Shared-everything OLTP **27**

3.1 Introduction 27

3.1.1 Multi-rooted B+Trees 28

3.1.2 Physiological Partitioning 29

3.1.3 Contributions and Organization 29

3.2 Communication Patterns 30

3.2.1 Types of Communication 30

3.2.2 Communication Patterns in OLTP 32

3.2.3 Physical vs. Logical Partitioning 33

3.3 Physiological Partitioning 34

3.3.1 Design Overview 34

3.3.2 Multi-rooted B+Tree 36

3.3.3 Heap Page Accesses 37

3.3.4 Page Cleaning 38

3.3.5 Benefits of Physiological Partitioning 38

3.4 Evaluation 40

3.4.1 Experimental Setup 40

3.4.2 Page Latches and Critical Sections 41

3.4.3 Reducing Index and Heap Page Latch Contention 43

3.4.4 Impact on Scalability and Performance 44

3.4.5 MRBTrees in Non-PLP Systems 46

3.4.6 Transactions with Joins in PLP 47

3.4.7 Secondary Index Accesses 48

3.4.8 Fragmentation Overhead 50

3.4.9 Summary 51

3.5 Related Work 52

3.5.1 Critical Sections 52

3.5.2 B+Trees and Alternative Concurrency Control 53

3.6 Limitations of PLP 54

3.7 PLP on Future Hardware and Conclusions 56

4 Dynamic Load Balancing for PLP **57**

4.1 Introduction 57

4.2 Need for Dynamic Repartitioning 59

4.3 Repartitioning Cost 60

4.3.1 Splitting Non-clustered Indexes 61

4.3.2 Splitting Clustered Indexes 65

4.3.3 Moving Fewer Records 65

4.3.4 Example of Repartitioning Cost 65

4.3.5 Cost of Merging Two Partitions 66

4.4	A Dynamic Load Balancing Mechanism for PLP	67
4.4.1	Monitoring	68
4.4.2	Deciding New Partitioning	70
4.4.3	Using Control Theory for Load Balancing	73
4.5	Evaluation	74
4.5.1	Experimental setup	74
4.5.2	Overhead in Normal Operation	75
4.5.3	Overhead of Updating Secondary Indexes for DLB	78
4.6	Related Work	79
4.7	Conclusions	80
II	Characterizing OLTP Benchmarks	81
5	From A to E: Analyzing TPC's OLTP Benchmarks	83
5.1	Introduction	83
5.2	Related Work	85
5.3	Setup and Methodology	86
5.3.1	Hardware	86
5.3.2	TPC-E Implementation	87
5.3.3	Software Setup	87
5.3.4	Experiments	88
5.4	Profiling Analysis	89
5.4.1	High-level Analysis	89
5.4.2	Time breakdown	90
5.5	Micro-architectural Analysis	92
5.5.1	OLTP on an Out-of-Order Processor	93
5.5.2	OLTP on an In-Order Processor	96
5.6	Summary of Results and Conclusion	98
6	Transactions under the Microscope	101
6.1	Introduction	101
6.2	Related Work	103
6.3	Setup and Methodology	104
6.4	Sensitivity to Data Size	106
6.5	Breakdown of Misses	107
6.5.1	Into Miss Categories	107
6.5.2	Into Operations	109
6.5.3	Into Components	110
6.6	Inside Transactions	111
6.6.1	Database Operations	111
6.6.2	Commonalities across Transactions	113
6.6.3	Average Reuse in an Instance	117

6.7	Conclusions	117
III	Chasing Instructions	119
7	Boosting Instruction Cache Reuse in OLTP	121
7.1	Introduction	122
7.2	Exploiting Instruction Overlap	124
7.3	Self-Assembly of Instruction Cache Collectives	125
7.3.1	SLICC Design	125
7.3.2	Implementation Requirements	128
7.3.3	Exploiting Transaction Type Information	130
7.3.4	Support for Thread Migration	131
7.4	Stratified Transaction Execution	131
7.4.1	STREX Synchronization Algorithm	132
7.4.2	Implementation	132
7.4.3	Effect on Regular Execution	133
7.5	Evaluation	134
7.5.1	Methodology	134
7.5.2	Exploring SLICC's Parameter Space	136
7.5.3	L1 Miss Rate	139
7.5.4	Throughput	140
7.5.5	Transaction Throughput vs. Latency	141
7.5.6	Hardware Cost	142
7.6	Related Work	144
7.7	Conclusions	145
8	Transaction-aware Instruction Chasing	147
8.1	Introduction	147
8.2	ADDICT	149
8.2.1	Finding Migration Points	149
8.2.2	Migrating Transactions	152
8.3	Evaluation	157
8.3.1	Setup and Methodology	157
8.3.2	Migration Points	159
8.3.3	Instruction and Data Misses	160
8.3.4	Performance Impact	161
8.3.5	Effect of Changing Loads	162
8.3.6	With Simultaneous Multithreading	163
8.3.7	On Deeper Memory Hierarchies	164
8.3.8	Overhead	165
8.3.9	Summary	166
8.4	Related Work	166

8.5	Conclusions	167
9	Future Directions and Concluding Remarks	169
9.1	Hardware Specialization	169
9.2	Other Applications to Benefit from Alternative Scheduling	170
9.3	Thesis Summary	172
	Bibliography	173
	Curriculum Vitae	189