



Hasso Plattner · Alexander Zeier

In-Memory Data Management

An Inflection Point
for Enterprise Applications

Buy this book on <http://amzn.com/3642193625>

 Springer

In-Memory Data Management

Hasso Plattner Alexander Zeier

In-Memory Data Management

An Inflection Point for Enterprise Applications

 Springer

Hasso Plattner
Alexander Zeier
Hasso Plattner Institute
Enterprise Platform and Integration Concepts
August-Bebel-Str. 88
14482 Potsdam
Germany
hasso.plattner@hpi.uni-potsdam.de
alexander.zeier@hpi.uni-potsdam.de

ISBN 978-3-642-19362-0 e-ISBN 978-3-642-19363-7
DOI 10.1007/978-3-642-19363-7
Springer Heidelberg Dordrecht London New York

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: WMX Design GmbH, Heidelberg, Germany

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

In Praise of *In-Memory Data Management: An Inflection Point for Enterprise Applications*

Academia

Prof. Christoph Meinel (Hasso Plattner Institute (HPI), Potsdam, Germany)

I'm proud that HPI and the cooperation between HPI and SAP has provided such an inspirational research environment that enabled the young research team around Hasso Plattner and Alexander Zeier to generate valuable and new scientific insights into the complex world of enterprise computations. Even more than that, they developed groundbreaking innovations that will open the door to a new age, the age in which managers can base their decisions on complex computational real-time analysis of business data, and thus will change the way how businesses are being operated.

Prof. David Simchi-Levi (Massachusetts Institute of Technology, Cambridge, USA)

This book describes a revolutionary database technology and many implementation examples for business intelligence and operations. Of particular interest to me are the opportunities opening up in supply chain management, where the need to balance the speed of planning algorithms with data granularity has been a long time obstacle to performance and usability.

Prof. Donald Kossmann (ETH Zurich, Switzerland)

This is the first book on in-memory database systems and how this technology can change the whole industry. The book describes how to build in-memory databases: what is different, what stays the same. Furthermore, the book describes how in-memory databases can become the single source of truth for a business.

Prof. Hector Garcia-Molina (Stanford University, California, USA)

Memory resident data can very significantly improve the performance of data intensive applications. This book presents an excellent overview of the issues and challenges related to in-memory data, and is highly recommended for anyone wishing to learn about this important area.

Prof. Hubert Oesterle (University of St. Gallen, Switzerland)

Technological innovations have again and again been enablers and drivers of innovative business solutions. As database management systems in the 1970s provided the grounds for ERP systems, which then enabled companies in almost all industries to redesign their business processes, upcoming in-memory databases will improve existing ERP-based business solutions (esp. in analytic processing) and will even lead to business processes and services being redesigned again. Plattner and Zeier describe the technical concepts of column- and row-based databases and encourage the reader to make use of the new technology in order to accomplish business innovation.

Prof. Michael Franklin (University of California, Berkeley, USA)

Hardware technology has evolved rapidly over the past decades, but database system architectures have not kept pace. At the same time, competition is forcing organizations to become more and more data-driven. These developments have driven a re-evaluation

of fundamental data management techniques and tradeoffs, leading to innovations that can exploit large memories, parallelism, and a deeper understanding of data management requirements. This book explains the powerful and important changes that are brought about by in-memory data processing. Furthermore, the unique combination of business and technological insights that the authors bring to bear provide lessons that extend beyond any particular technology, serving as a guidebook for innovation in this and future Information Technology revolutions.

Prof. Sam Madden (Massachusetts Institute of Technology, Cambridge, USA)

Plattner and Zeier's book is a thorough accounting of the need for, and the design of, main memory database systems. By analyzing technology trends, they make a compelling case for the coming dominance of main-memory in database systems. They go on to identify a series of key design elements that main memory database system should have, including a column-oriented design, support for multi-core processor parallelism, and data compression. They also highlight several important requirements imposed by modern business processes, including heavy use of stored procedures and accounting requirements that drive a need for no-overwrite storage. This is the first book of it's kind, and it provides a complete reference for students and database designers alike.

Prof. Terry Winograd (Stanford University, California, USA)

There are moments in the development of computer technology when the ongoing evolution of devices changes the tradeoffs to allow a tectonic shift – a radical change in the way we interact with computers. The personal computer, the Web, and the smart phone are all examples where long-term trends reached a tipping point allowing explosive change and growth. Plattner and Zeier present a vision of how this kind of radical shift is coming to enterprise data management. From Plattner's many years of executive experience and development of data management systems, he is able to see the new space of opportunities for users – the potential for a new kind of software to provide managers with a powerful new tool for gaining insight into the workings of an enterprise. Just as the web and the modern search engine changed our idea of how, why, and when we “retrieve information,” large in-memory databases will change our idea of how to organize and use operational data of every kind in every enterprise. In this visionary and valuable book, Plattner and Zeier lay out the path for the future of business.

Prof. Warren B. Powell (Princeton University, Princeton, New Jersey, USA)

In this remarkable book, Plattner and Zeier propose a paradigm shift in memory management for modern information systems. While this offers immediate benefits for the storage and retrieval of images, transaction histories and detailed snapshots of people, equipment and products, it is perhaps even more exciting to think of the opportunities that this technology will create for the future. Imagine the fluid graphical display of spatially distributed, dynamic information. Or the ability to move past the flat summaries of inventories of equipment and customer requests to capture the subtle context that communicates urgency and capability. Even more dramatic, we can envision the real-time optimization of business processes working interactively with domain experts, giving us the information-age equivalent of the robots that make our cars and computers in the physical world today.

Prof. Wolfgang Lehner (Technical University of Dresden, Germany)

This book shows in an extraordinary way how technology can drive new applications – a fascinating journey from the core characteristics of business applications to topics of leading-edge main-memory database technology.

Industry

Bill McDermott (Co-CEO, SAP, Newtown Square, Pennsylvania, USA)

We are witnessing the dawn of a new era in enterprise business computing, defined by the near instantaneous availability of critical information that will drive faster decision making, new levels of business agility, and incredible personal productivity for business users. With the advent of in-memory technology, the promise of real-time computing is now reality, creating a new inflection point in the role IT plays in driving sustainable business value. In their review of in-memory technology, Hasso Plattner and Alexander Zeier articulate how in-memory technology can drive down costs, accelerate business, help companies reap additional value out of their existing IT investments, and open the door to new possibilities in how business applications can be consumed. This work is a “must read” for anyone who leverages IT innovation for competitive advantage.

Falk F. Strascheg (Founder and General Partner, EXTOREL, Munich, Germany)

Since the advent of the Internet we have been witnessing new technologies coming up quickly and frequently. It is however rare that these technologies become innovations in the sense that there are big enough market opportunities. Hasso Plattner has proven his ability to match business needs with technical solutions more than once, and this time he presents the perhaps most significant innovation he has ever been working on: Real-Time Business powered by In-Memory Computing. As the ability for innovation has always been one of the core factors for competitiveness this is a highly advisable piece of reading for all those who aim to be at the cutting edge.

Gerhard Oswald (COO, SAP, Walldorf, Germany)

In my role as COO of SAP it is extremely important to react quickly to events and to have instant access to the current state of the business. At SAP, we have already moved a couple of processes to the new in-memory technology described in the book by Hasso Plattner and Alexander Zeier. I’m very excited about the recently achieved improvements utilizing the concepts described in this book. For example, I monitor our customer support messaging system every day using in-memory technology to make sure that we provide our customers with the timely responses they deserve. I like that this book provides an outlook of how companies can smoothly adopt the new database technology. This transition concept, called the bypass solution, gives our existing customer base the opportunity to benefit from this fascinating technology, even for older releases of SAP software.

Hermann-Josef Lamberti (COO, Deutsche Bank, Frankfurt, Germany)

Deutsche Bank has run a prototype with an early versions of the in-memory technology described in the book by Hasso Plattner and Alexander Zeier. In particular, we were able to speed up the data analysis process to detect cross-selling opportunities in our customer database, from previously 45 minutes to 5 seconds. In-memory is a powerful new dimension of applied compute power.

Jim Watson (Managing General Partner, CMEA Capital, San Francisco, California, USA)

During the last 50 years, every IT era has brought us a major substantial advancement, ranging from mainframe computers to cloud infrastructures and smart-phones. In certain decades the strategic importance of one technology versus the other is dramatically different and it may fundamentally change the way in which people do business. This is what a Venture Capitalist has to bear in mind when identifying new trends that are along for the long haul. In their book, Hasso and Alex do not only describe a market-driven innovation from Germany, that has the

potential to change the enterprise software market as a whole, but they also present a working prototype.

Martin Petry (CIO, Hilti, Schaan, Liechtenstein)

Hilti is a very early adopter of the in-memory technology described in the book by Hasso Plattner and Alexander Zeier. Together with SAP, we have worked on developing prototypical new applications using in-memory technology. By merging the transactional world with the analytical world these applications will allow us to gain real-time insight into our operations and allow us to use this insight in our interaction with customers. The benefit for Hilti applying SAP's in-memory technology is not only seen in a dramatic improvement of reporting execution speed - for example, we were able to speed up a reporting batch job from 3 hours to seconds – but even more in the opportunity to bring the way we work with information and ultimately how we service our customers on a new level.

Prof. Norbert Walter (former Chief Economist of Deutsche Bank, Frankfurt, Germany)

Imagine you feel hungry. But instead of just opening the fridge (imagine you don't have one) to get hold of, say, some butter and cheese, you would have to leave the house for the nearest dairy farm. Each time you feel hungry. This is what we do today with most company data: We keep them far away from where we process them. In their highly accessible book, Hasso Plattner and Alexander Zeier show how in-memory technology moves data where they belong, promising massive productivity gains for the modern firm. Decision makers, get up to speed!

Paul Polman (CEO, Unilever; London, UK)

There are big opportunities right across our value chain to use real time information more imaginatively. Deeper, real time insight into consumer and shopper behavior will allow us to work even more closely and effectively with our customers, meeting the needs of today's consumers. It will also transform the way in which we serve our customers and consumers and the speed with which we do it. I am therefore very excited about the potential that the in-memory database technology offers to my business.

Tom Greene (CIO, Colgate-Palmolive Company, New York City, USA)

In their book, Hasso Plattner and Alexander Zeier do not only describe the technical foundations of the new data processing capabilities coming from in-memory, but they also provide examples for new applications that can now be built on top. For a company like Colgate-Palmolive, these new applications are of strategic importance, as they allow for new ways of analyzing our transactional data in real time, which can give us a competitive advantage.

Dr. Vishal Sikka (CTO, Executive Board Member; SAP, Palo Alto, California, USA)

Hasso Plattner is not only an amazing entrepreneur, he is an incredible teacher. His work and his teaching have inspired two generations of students, leaders, professionals and entrepreneurs. Over the last five years, we have been on a fantastic journey with him, from his early ideas on rethinking our core financials applications, to conceiving and implementing a completely new data management foundation for all our SAP products. This book by Hasso and Alexander, captures these experiences and I encourage everyone in enterprise IT to read this book and take advantage of these learnings, just as I have endeavored to embody these in our products at SAP.

To Annabelle and my family

AZ

Foreword

By

Prof. John L. Hennessy (Stanford University, California, USA) and

Prof. David A. Patterson (University of California at Berkeley, USA)

Is anyone else in the world both as well-qualified as Hasso Plattner to make a strong business case for real-time data analytics *and* describe the technical details for a solution based on insights in database design for Enterprise Resource Planning that leverage recent hardware technology trends?

The P of SAP has been both the CEO of a major corporation and a Professor of Computer Science at a leading research institute, where he and his colleagues built a working prototype of a main memory database for ERP, proving once again that Hasso Plattner is a person who puts his full force into the things he believes in. Taking advantage of rapid increases in DRAM capacity and in the number of the processors per chip, SanssouciDB demonstrates that the traditional split of separate systems for Online Transaction Processing (OLTP) and for Online Analytical Processing (OLAP) is no longer necessary for ERP.

Business leaders now can ask ad hoc questions of the production transaction database and get the answer back in seconds. With the traditional divided OLTP/OLAP systems, it can take a week to write the query and receive the answer. In addition to showing how software can use concepts from shared nothing databases to scale across blade servers and use concepts from shared everything databases to take advantage of the large memory and many processors inside a single blade, this book touches on the role of Cloud Computing to achieve a single system for transactions and analytics.

Equally as important as the technical achievement, the “Bill Gates of Germany” shows how businesses can integrate this newfound ability to improve the efficiency and profitability of business, and in a time when so many businesses are struggling to deal with the global problem of markets and supply chains, this instant analytical ability could not be more important. Moreover, if this ability is embraced and widely used, perhaps business leaders can quickly and finely adjust enterprise resources to meet rapidly varying demands so that the next economic downturn will not be as devastating to the world’s economy as the last one.

Preface

We wrote this book because we think that the use of in-memory technology marks an inflection point for enterprise applications. The capacity per dollar and the availability of main memory has increased markedly in the last few years. This has led to a rethinking of how mass data should be stored. Instead of using mechanical disk drives it is now possible to store the primary data copy of a database in silicon-based main memory resulting in an orders-of-magnitude improvement in performance and allowing completely new applications to be developed. This change in the way data is stored is having, and will continue to have a significant impact on enterprise applications and ultimately on the way businesses are run. Having real-time information available at the speed of thought provides decision makers in an organization with insights that have, until now, not existed.

This book serves the interests of specific reader groups. Generally, the book is intended for anyone who wishes to find out how this fundamental shift in the way data is managed is affecting, and will continue to affect enterprise applications. In particular, we hope that university students, IT professionals and IT managers, as well as senior management, who wish to create new business processes by leveraging in-memory computing, will find this book inspiring.

The book is divided into three parts:

- *Part I* gives an overview of our vision of how in-memory technology will change enterprise applications. This part will be of interest to all readers.
- *Part II* provides a more in-depth description of how we intend to realize our vision, and addresses students and developers, who want a deeper technical understanding of in-memory data management.
- *Part III* describes the resulting implications on the development and capabilities of enterprise applications, and is suited for technical as well as business-oriented readers.

Writing a book always involves more people than just the authors. We would like to thank the members of our Enterprise Platform and Integration Concepts group at the Hasso Plattner Institute at the University of Potsdam in Germany. Anja Bog, Martin Grund, Jens Krüger, Stephan Müller, Jan Schaffner, and Christian Tinnefeld are part of the HANA research group and their work over the last five years in the field of in-memory applications is the foundation for our book. Vadym Borovskiy, Thomas Kowark, Ralph Kühne, Martin Lorenz, Jürgen Müller, Oleksandr Panchenko, Matthieu Schapranow, Christian Schwarz, Matthias Uflacker, and Johannes Wust

also made significant contributions to the book and our assistant Andrea Lange helped with the necessary coordination.

Additionally, writing this book would not have been possible without the help of many colleagues at SAP. Cafer Tosun in his role as the link between HPI and SAP not only coordinates our partnership with SAP, but also actively provided sections for our book. His team members Andrew McCormick-Smith and Christian Mathis added important text passages to the book. We are grateful for the work of Joos-Hendrik Böse, Enno Folkerts, Sarah Kappes, Christian Münkel, Frank Renkes, Frederik Transier, and other members of his team. We would like to thank Paul Hofmann for his input and for his valuable help managing our research projects with American universities. The results we achieved in our research efforts would also not have been possible without the outstanding help of many other colleagues at SAP. We would particularly like to thank Franz Färber and his team for their feedback and their outstanding contributions to our research results over the past years. Many ideas that we describe throughout the book were originally Franz's, and he is also responsible for their implementation within SAP. We especially want to emphasize his efforts.

Finally, we want to express our gratitude to SAP CTO Vishal Sikka for his sponsorship of our research and his personal involvement in our work. In addition, we are grateful to SAP COO Gerhard Oswald and SAP Co-CEOs Jim Hagemann Snabe and Bill McDermott for their ongoing support of our projects.

We encourage you to visit the official website of this book. The website contains updates about the book, reviews, blog entries about in-memory data management, and examination questions for students. You can access the book's website via:

no-disk.com

Contents

Foreword	XI
<i>By</i> <i>Prof. John L. Hennessy (Stanford University, California, USA) and</i> <i>Prof. David A. Patterson (University of California at Berkeley, USA)</i>	
Preface	XIII
Introduction	1
PART I – An Inflection Point for Enterprise Applications	5
1 Desirability, Feasibility, Viability – The Impact of In-Memory	7
1.1 Information in Real Time – Anything, Anytime, Anywhere	7
1.1.1 Response Time at the Speed of Thought	9
1.1.2 Real-Time Analytics and Computation on the Fly	10
1.2 The Impact of Recent Hardware Trends	11
1.2.1 Database Management Systems for Enterprise Applications ..	11
1.2.2 Main Memory Is the New Disk	14
1.2.3 From Maximizing CPU Speed to Multi-Core Processors	15
1.2.4 Increased Bandwidth between CPU and Main Memory	17
1.3 Reducing Cost through In-Memory Data Management	20
1.3.1 Total Cost of Ownership	20
1.3.2 Cost Factors in Enterprise Systems	21
1.3.3 In-Memory Performance Boosts Cost Reduction	22
1.4 Conclusion	23
2 Why Are Enterprise Applications So Diverse?	25
2.1 Current Enterprise Applications	25
2.2 Examples of Enterprise Applications	27
2.3 Enterprise Application Architecture	29
2.4 Data Processing in Enterprise Applications	30
2.5 Data Access Patterns in Enterprise Applications	31
2.6 Conclusion	31

- 3 SanssouciDB – Blueprint for an In-Memory Enterprise Database System** 33
 - 3.1 Targeting Multi-Core and Main Memory 34
 - 3.2 Designing an In-Memory Database System 36
 - 3.3 Organizing and Accessing Data in Main Memory 37
 - 3.4 Conclusion 40

- PART II – SanssouciDB – A Single Source of Truth through In-Memory . .** 41

- 4 The Technical Foundations of SanssouciDB** 43
 - 4.1 Understanding Memory Hierarchies 43
 - 4.1.1 Introduction to Main Memory 44
 - 4.1.2 Organization of the Memory Hierarchy 47
 - 4.1.3 Trends in Memory Hierarchies 49
 - 4.1.4 Memory from a Programmer’s Point of View 50
 - 4.2 Parallel Data Processing Using Multi-Core and Across Servers 57
 - 4.2.1 Increasing Capacity by Adding Resources 57
 - 4.2.2 Parallel System Architectures 59
 - 4.2.3 Parallelization in Databases for Enterprise Applications 61
 - 4.2.4 Parallel Data Processing in SanssouciDB 64
 - 4.3 Compression for Speed and Memory Consumption 68
 - 4.3.1 Light-Weight Compression 69
 - 4.3.2 Heavy-Weight Compression 73
 - 4.3.3 Data-Dependent Optimization 73
 - 4.3.4 Compression-Aware Query Execution 73
 - 4.3.5 Compression Analysis on Real Data 74
 - 4.4 Column, Row, Hybrid – Optimizing the Data Layout. 75
 - 4.4.1 Vertical Partitioning 75
 - 4.4.2 Finding the Best Layout 78
 - 4.4.3 Challenges for Hybrid Databases 81
 - 4.5 The Impact of Virtualization 81
 - 4.5.1 Virtualizing Analytical Workloads 82
 - 4.5.2 Data Model and Benchmarking Environment 82
 - 4.5.3 Virtual versus Native Execution 83
 - 4.5.4 Response Time Degradation with Concurrent VMs 84
 - 4.6 Conclusion 86

- 5 Organizing and Accessing Data in SanssouciDB** 89
 - 5.1 SQL for Accessing In-Memory Data 90
 - 5.1.1 The Role of SQL 90
 - 5.1.2 The Lifecycle of a Query 91
 - 5.1.3 Stored Procedures 91
 - 5.1.4 Data Organization and Indices 91
 - 5.2 Increasing Performance with Data Aging 92
 - 5.2.1 Active and Passive Data 93

- 5.2.2 Implementation Considerations for an Aging Process 95
- 5.2.3 The Use Case for Horizontal Partitioning of Leads 95
- 5.3 Efficient Retrieval of Business Objects 98
 - 5.3.1 Retrieving Business Data from a Database 98
 - 5.3.2 Object Data Guide 99
- 5.4 Handling Data Changes in Read-Optimized Databases 100
 - 5.4.1 The Impact on SanssouciDB 101
 - 5.4.2 The Merge Process 103
 - 5.4.3 Improving Performance with Single Column Merge 107
- 5.5 Append, Never Delete, to Keep the History Complete 109
 - 5.5.1 Insert-Only Implementation Strategies 110
 - 5.5.2 Minimizing Locking through Insert-Only 111
 - 5.5.3 The Impact on Enterprise Applications 114
 - 5.5.4 Feasibility of the Insert-Only Approach 117
- 5.6 Enabling Analytics on Transactional Data 118
 - 5.6.1 Aggregation on the Fly 119
 - 5.6.2 Analytical Queries without a Star Schema 128
- 5.7 Extending Data Layout without Downtime 135
 - 5.7.1 Reorganization in a Row Store 135
 - 5.7.2 On-The-Fly Addition in a Column Store 136
- 5.8 Business Resilience through Advanced Logging Techniques 137
 - 5.8.1 Recovery in Column Stores 138
 - 5.8.2 Differential Logging for Row-Oriented Databases 140
 - 5.8.3 Providing High Availability 141
- 5.9 The Importance of Optimal Scheduling for Mixed Workloads 142
 - 5.9.1 Introduction to Scheduling 142
 - 5.9.2 Characteristics of a Mixed Workload 145
 - 5.9.3 Scheduling Short and Long Running Tasks 146
- 5.10 Conclusion 148

PART III – How In-Memory Changes the Game 151

- 6 Application Development 153**
 - 6.1 Optimizing Application Development for SanssouciDB 153
 - 6.1.1 Application Architecture 154
 - 6.1.2 Moving Business Logic into the Database 155
 - 6.1.3 Best Practices 157
 - 6.2 Innovative Enterprise Applications 158
 - 6.2.1 New Analytical Applications 158
 - 6.2.2 Operational Processing to Simplify Daily Business 162
 - 6.2.3 Information at Your Fingertips with Innovative User-Interfaces 164
 - 6.3 Conclusion 169

- 7 Finally, a Real Business Intelligence System Is at Hand** 171
 - 7.1 Analytics on Operational Data 171
 - 7.1.1 Yesterday’s Business Intelligence 171
 - 7.1.2 Today’s Business Intelligence 174
 - 7.1.3 Drawbacks of Separating Analytics from Daily Operations . . . 176
 - 7.1.4 Dedicated Database Designs for Analytical Systems 178
 - 7.1.5 Analytics and Query Languages 180
 - 7.1.6 Enablers for Changing Business Intelligence 182
 - 7.1.7 Tomorrow’s Business Intelligence 183
 - 7.2 How to Evaluate Databases after the Game Has Changed 185
 - 7.2.1 Benchmarks in Enterprise Computing 185
 - 7.2.2 Changed Benchmark Requirements for a Mixed Workload . . . 187
 - 7.2.3 A New Benchmark for Daily Operations and Analytics 188
 - 7.3 Conclusion 192

- 8 Scaling SanssouciDB in the Cloud** 193
 - 8.1 What Is Cloud Computing? 194
 - 8.2 Types of Cloud Applications 195
 - 8.3 Cloud Computing from the Provider Perspective 197
 - 8.3.1 Multi-Tenancy 197
 - 8.3.2 Low-End versus High-End Hardware 201
 - 8.3.3 Replication 201
 - 8.3.4 Energy Efficiency by Employing In-Memory Technology 202
 - 8.4 Conclusion 204

- 9 The In-Memory Revolution Has Begun** 205
 - 9.1 Risk-Free Transition to In-Memory Data Management 205
 - 9.1.1 Operating In-Memory and Traditional Systems Side by Side . . 206
 - 9.1.2 System Consolidation and Extensibility 207
 - 9.2 Customer Proof Points 208
 - 9.3 Conclusion 209

- References** 211

- About the Authors** 221

- Glossary** 223

- Abbreviations** 231

- Index** 233

Introduction

Over the last 50 years, advances in Information Technology (IT) have had a significant impact on the success of companies across all industries. The foundations for this success are the interdependencies between business and IT, as they not only address and ease the processing of repetitive tasks, but are the enabler for creating more accurate and complete insights into a company. This aspect has often been described and associated with the term real-time as it suggests that every change that happens within a company is instantly visible through IT.

We think that significant milestones have been reached towards this goal throughout the history of enterprise computing, but we are not there, yet. Currently, most of the data within a company is still distributed throughout a wide range of applications and stored in several disjoint silos. Creating a unified view on this data is a cumbersome and time-consuming procedure. Additionally, analytical reports typically do not run directly on operational data, but on aggregated data from a data warehouse. Operational data is transferred into this data warehouse in batch jobs, which makes flexible, ad-hoc reporting on up-to-date data almost impossible. As a consequence, company leaders have to make decisions based on insufficient information, which is not what the term real-time suggests. We predict this is about to change as hardware architectures have evolved dramatically in the last decade. Multi-core processors and the availability of large amounts of main memory at low cost are creating new breakthroughs in the software industry. It has become possible to store data sets of whole companies entirely in main memory, which offers performance that is orders of magnitudes faster than traditional disk-based systems. Hard disks will become obsolete. The only remaining mechanical device in a world of silicon will soon only be necessary for backing up data. With in-memory computing and insert-only databases using row- and column-oriented storage, transactional and analytical processing can be unified. High performance in-memory computing will change how enterprises work and finally offer the promise of real-time computing.

As summarized in Figure I, the combination of the technologies mentioned above finally enables an iterative link between the instant analysis of data, the prediction of business trends, and the execution of business decisions without delays.

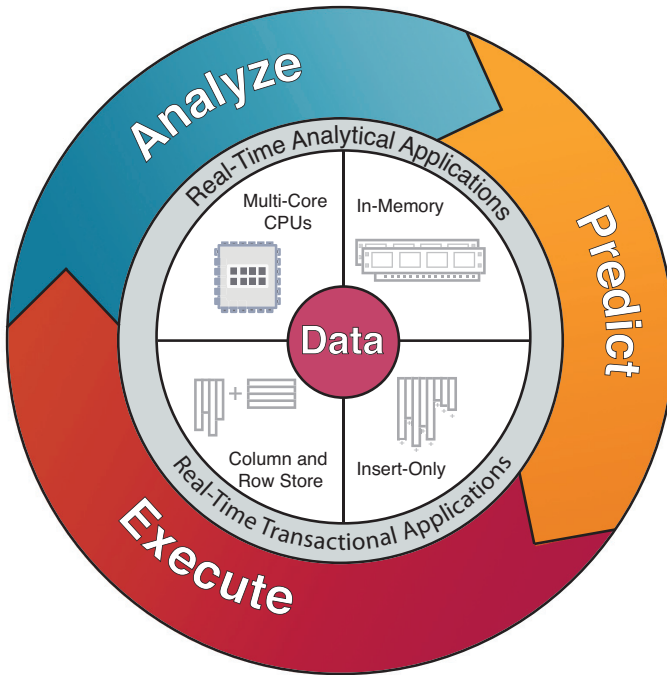


Figure I: Enterprise Performance In-Memory Circle (EPIC)

How can companies take advantage of in-memory applications to improve the efficiency and profitability of their business? We predict that this break-through innovation will lead to fundamentally improved business processes, better decision-making, and new performance standards for enterprise applications across industries and organizational hierarchies. We are convinced that in-memory technology is a catalyst for innovation, and the enabler for a level of information quality that has not been possible until now. In-memory enterprise data management provides the necessary equipment to excel in a future where businesses face ever-growing demands from customers, partners, and shareholders. With billions of users and a hundred times as many sensors and devices on the Internet, the amount of data we are confronted with is growing exponentially. Being able to quickly extract business-relevant information not only provides unique opportunities for businesses; it will be a critical differentiator in future competitive markets.

With in-memory technology, companies can fully leverage massive amounts of data to create strategic advantage. Operational business data can be interactively analyzed and queried without support from the IT department, opening up completely new scenarios and opportunities.

Consider financial accounting, where data needs to be frequently aggregated for reporting on a daily, weekly, monthly, or annual basis. With in-memory data management, the necessary filtering and aggregation can happen in real time. Accounting can be done anytime and in an ad-hoc manner. Financial applications

will not only be significantly faster, they will also be less complex and easier to use. Every user of the system will be able to directly analyze massive amounts of data. New data is available for analysis as soon as it is entered into the operational system. Simulations, forecasts, and what-if scenarios can be done on demand, anytime and anywhere. What took days or weeks in traditional disk-based systems can now happen in the blink of an eye. Users of in-memory enterprise systems will be more productive and responsive.

The concepts presented in this book create new opportunities and improvements across all industries. Below, we present a few examples:

- *Daily Operations*: Gain real-time insight into daily revenue, margin, and labor expenses.
- *Competitive Pricing*: Intuitively explore impact of competition on product pricing to instantly understand impact to profit contribution.
- *Risk Management*: Immediately identify high-risk areas across multiple products and services and run what-if scenario analyses on the fly.
- *Brand and Category Performance*: Evaluate the distribution and revenue performance of brands and product categories by customer, region, and channel at any time.
- *Product Lifecycle and Cost Management*: Get immediate insight into yield performance versus customer demand.
- *Inventory Management*: Optimize inventory and reduce out-of-stocks based on live business events.
- *Financial Asset Management*: Gain a more up-to-date picture of financial markets to manage exposure to currencies, equities, derivatives, and other instruments.
- *Real-Time Warranty and Defect Analysis*: Get live insight into defective products to identify deviation in production processes or handling.

In summary, we foresee in-memory technology triggering the following improvements in the following three interrelated strategic areas:

- *Reduced Total Cost of Ownership*: With our in-memory data management concepts, the required analytical capabilities are directly incorporated into the operational enterprise systems. Dedicated analytical systems are a thing of the past. Enterprise systems will become less complex and easier to maintain, resulting in less hardware maintenance and IT resource requirements.
- *Innovative Applications*: In-memory data management combines high-volume transactions with analytics in the operational system. Planning, forecasting, pricing optimization, and other processes can be dramatically improved and supported with new applications that were not possible before.
- *Better and Faster Decisions*: In-memory enterprise systems allow quick and easy access to information that decision makers need, providing them with new ways to look at the business. Simulation, what-if analyses, and planning can be performed interactively on operational data. Relevant information is instantly accessible and the reliance on IT resources is reduced. Collaboration within and across organizations is simplified and fostered. This can lead to a

much more dynamic management style where problems can be dealt with as they happen.

At the research group Enterprise Platform and Integration Concepts under the supervision of Prof. Dr. Hasso Plattner and Dr. Alexander Zeier at the Hasso Plattner Institute (HPI) we have been working since 2006 on research projects with the goal of revolutionizing enterprise systems and applications. Our vision is that in-memory computing will enable completely new ways of operating a business and fulfill the promise of real-time data processing. This book serves to explain in-memory database technology and how it is an enabler for this vision. We go on to describe how this will change the way enterprise applications are developed and used from now on.

PART I – An Inflection Point for Enterprise Applications

For as long as businesses have existed, decision makers have wanted to know the current state of their company. As businesses grow, however, working out exactly where the money, materials, and people go becomes more and more complicated. Tools are required to help to keep track of everything. Since the 1960s, computers have been used to perform this task and complex software systems called enterprise applications have been created to provide insights into the daily operations of a company. However, increasing data volumes have meant that by the turn of the 21st century, large organizations were no longer always able to access the information they required in a timely manner.

At the heart of any enterprise application is the database management system, responsible for storing the myriad of data generated by the day-to-day operations of a business. In the first part of this book, we provide a brief introduction to enterprise applications and the databases that underlie them. We also introduce the technology that we believe has created an inflection point in the development of these applications. In Chapter 1 we explain the desirability, feasibility, and viability of in-memory data management. Chapter 2 introduces the complexity and common data access patterns of enterprise applications. Chapter 3 closes the first part with the description of SanssouciDB, our prototypical in-memory database management system.

1 Desirability, Feasibility, Viability – The Impact of In-Memory

Abstract Sub-second response time and real-time analytics are key requirements for applications that allow natural human computer interactions. We envision users of enterprise applications to interact with their software tools in such a natural way, just like any Internet user interacts with a web search engine today by refining search results on the fly when the initial results are not satisfying. In this initial chapter, we illustrate this vision of providing business data in real time and discuss it in terms of desirability, feasibility, and viability. We first explain the desire of supplying information in real time and review sub-second response time in the context of enterprise applications. We then discuss the feasibility based on in-memory databases that leverage modern computer hardware and conclude by demonstrating the economic viability of in-memory data management.

In-memory technology is set to revolutionize enterprise applications both in terms of functionality and cost due to a vastly improved performance. This will enable enterprise developers to create completely new applications and allow enterprise users and administrators to think in new ways about how they wish to view and store their data. The performance improvements also mean that costly workarounds, necessary in the past to ensure data could be processed in a timely manner, will no longer be necessary. Chief amongst these is the need for separate operational and analytical systems. In-memory technology will allow analytics to be run on operational data, simplifying both the software and the hardware landscape, leading ultimately to lower overall cost.

1.1 Information in Real Time – Anything, Anytime, Anywhere

Today's web search engines show us the potential of being able to analyze massive amounts of data in real time. Users enter their queries and instantly receive answers. The goal of enterprise applications in this regard is the same, but is barely reached. For example, call center agents or managers are looking for specific pieces of information within all data sources of the company to better decide on products to offer to customers or to plan future strategies. Compared to web search with

its instant query results, enterprise applications are slower, exposing users to noticeably long response times. The behavior of business users would certainly change if information was as instantly available in the business context as in the case of web search.

One major difference between web search and enterprise applications is the completeness of the expected results. In a web search only the hits that are rated most relevant are of interest, whereas all data relevant for a report must be scanned and reflected in its result. A web search sifts through an indexed set of data evaluating relevance and extracting results. In contrast, enterprise applications have to do additional data processing, such as complex aggregations. In a number of application scenarios, such as analytics or planning, data must be prepared before it is ready to be presented to the user, especially if the data comes from different source systems.

Current operational and analytical systems are separated to provide the ability to analyze enterprise data and to reach adequate query response times. The data preparation for analytics is applied to only a subset of the entire enterprise data set. This limits the data granularity of possible reports. Depending on the steps of preparation, for example, data cleansing, formatting, or calculations, the time window between data being entered into the operational system until being available for reporting might stretch over several hours or even days (see Section 7.1 for a more detailed discussion of the reasons, advantages, and drawbacks of the separation). This delay has a particular effect on performance when applications need to do both operational processing and analytics. Available-to-Promise (ATP), demand planning, and dunning applications introduced in Chapter 2 are examples of these types of applications. They show characteristics associated with operational processing as they must operate on up-to-date data and perform read and write operations. They also reveal characteristics that are associated with analytics like processing large amounts of data because recent and historical data is required. These applications could all benefit from the ability to run interactive what-if scenarios. At present, sub-second response times in combination with the flexible access to any information in the system are not available.

Figure 1.1 is an interpretation of information at the fingertips; a term coined by Bill Gates in 1994, when he envisioned a future in which arbitrary information is available from anywhere [1]. Our interpretation shows meeting attendees situated in several locations, all browsing, querying, and manipulating the same information in real time. The process of exchanging information can be shortened while being enriched with the potential to include and directly answer ad-hoc queries.

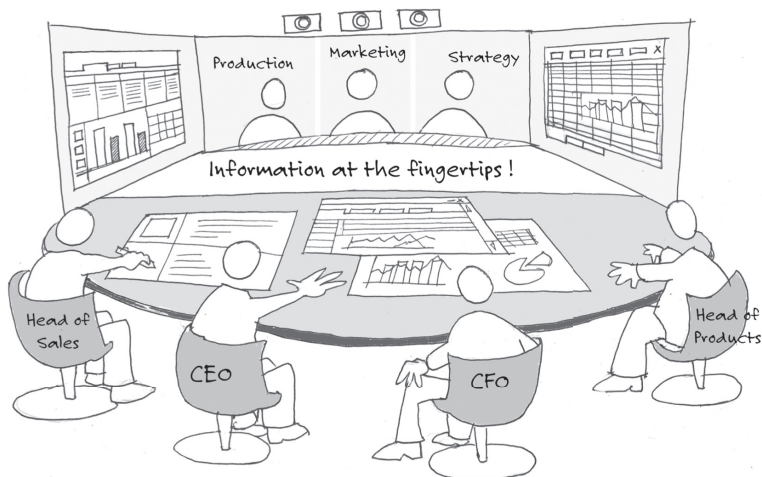


Figure 1.1: Management Meeting of the Future [2]

We now expand on the topics of sub-second response time, real-time analytics, and computation on the fly. These topics are vital for the scenario sketched above.

1.1.1 Response Time at the Speed of Thought

In web search, users query data using key word search. The meaning of key words may be ambiguous, which results in users redefining their search terms according to the received results. Sub-second response time is the enabler of such trial-and-error behavior. With sub-second response time in enterprise analytics, users would be able to employ the same method to query business data. Reporting queries could be defined and redefined interactively without long waiting periods to prepare data or create new reports.

The average simple reaction time for an observer to form a simple response to the presence of a stimulus is 220 ms [3]. Part of this time is needed to detect the stimulus and the remainder to organize the response. Recognition reaction time is longer because the process of understanding and comprehension has to take place. The average recognition reaction time has been measured to be 384 ms. Depending on the complexity of the context, the recognition reaction time increases. Assuming more complex contexts, the interval of 550 to 750 ms is what we call speed of thought. For trained users, who repeatedly perform the same action, the reaction times can be shorter than the above numbers and slow system response times will seem even longer for them.

Any interval sufficiently longer than the speed-of-thought interval will be detected as waiting time and the user's mind starts wandering to other topics, which is a process that cannot be consciously controlled. The longer the interval, the further the mind is taken off the task at hand. Sub-second response time is one step into the direction of helping the user to focus on a topic and not to become distracted

by other tasks during waiting periods. Context switching between tasks is extremely tiring. Even for small tasks, the user has to find his way back to the topic and needs to remember what the next step was. If such a context switch can be omitted, the user's full attention is dedicated to exploring and analyzing data. The freedom to build queries on the results of previous ones without becoming distracted helps the user to dig deeper into data in a much shorter time.

Sub-second response time means that we can use enterprise applications in completely new ways, for example, on mobile devices. The expectation of mobile device users is that they will not have to wait more than a few seconds for a response from their device. If we are able to get a sub-second response time from our enterprise application then the response time for the mobile user, including the transfer time, will be within acceptable limits. This would allow managers to get the results of a dunning run on their phone while waiting for a flight. They could then call the worst debtors directly, hopefully leading to a rapid resolution of the problem. Compare this with traditional systems where a dunning run can potentially take hours.

1.1.2 Real-Time Analytics and Computation on the Fly

The basis for real-time analytics is to have all resources at disposal in the moment they are called for [4]. So far, special materialized data structures, called cubes, have been created to efficiently serve analytical reports. Such cubes are based on a fixed number of dimensions along which analytical reports can define their result sets. Consequently, only a particular set of reports can be served by one cube. If other dimensions are needed, a new cube has to be created or existing ones have to be extended. In the worst case, a linear increase in the number of dimensions of a cube can result in an exponential growth of its storage requirements. Extending a cube can result in a deteriorating performance of those reports already using it. The decision to extend a cube or build a new one has to be considered carefully. In any case, a wide variety of cubes may be built during the lifetime of a system to serve reporting, thus increasing storage requirements and also maintenance efforts.

Instead of working with a predefined set of reports, business users should be able to formulate ad-hoc reports. Their playground should be the entire set of data the company owns, possibly including further data from external sources. Assuming a fast in-memory database, no more pre-computed materialized data structures are needed. As soon as changes to data are committed to the database, they will be visible for reporting. The preparation and conversion steps of data if still needed for reports are done during query execution and computations take place on the fly. Computation on the fly during reporting on the basis of cubes that do not store data, but only provide the interface for reporting, solves a problem that has existed up to now and allows for performance optimization of all analytical reports likewise.

1.2 The Impact of Recent Hardware Trends

Modern hardware is subject to continuous change and innovation, of which the most recent are the emergence of multi-core architectures and larger, less expensive main memory.¹ Existing software systems, such as database management systems, must be adapted to keep pace with these developments and exploit, to the maximum degree, the potential of the underlying hardware. In this section, we introduce database management systems and outline recent trends in hardware. We point out why those are key enablers for in-memory data management which makes true real-time analytics in the context of enterprise applications feasible.

1.2.1 Database Management Systems for Enterprise Applications

We define a Database Management System (DBMS) as a collection of programs that enable users to create and maintain a database [5]. The DBMS is a software system that facilitates the process of defining, constructing, manipulating, and sharing databases among various users and applications. It underpins all operations in an enterprise application and the performance of the enterprise application as a whole is heavily dependent on the performance of the DBMS.

Improving the performance of the database layer is a key aspect of our goal to remove the need for separate analytical systems and thus allow real-time access to all the data in an enterprise system. Here we describe the database concepts that are most relevant to enterprise applications and how hardware limitations at the database layer drove the decision to split Business Intelligence (BI) applications from the transactional system. We also discuss how recent developments in hardware are making it possible to keep all of an enterprise application's data in main memory, resulting in significant performance gains.

The requirements of enterprise applications have been a key driver in the development of DBMSs, both in the academic and commercial worlds. Indeed, one of the earliest commercial DBMSs was the Information Management System (IMS) [6], developed in 1968 by IBM for inventory management in the National Aeronautics and Space Administration's Apollo space program. IMS was typical of the first commercial database systems in that it was built for a particular application.² At that time the effort required to develop complex software systems [7] meant that a DBMS that could provide a variety of general purpose functions to an application layer had not yet been developed. Creating new DBMSs to address each new enterprise application's needs was obviously time-consuming and inefficient; the motivation to have DBMSs that could support a wide range of

¹ Main memory refers to silicon-based storage directly accessible from the CPU while in-memory refers to the concept of storing the primary data copy of a database in main memory. Main memory is volatile as data is lost upon power failure.

² It should be noted that reengineered versions of IMS have subsequently come to be used in a wide variety of scenarios [6].

enterprise applications was strong. Their development became a common goal of both research and industry.

Enterprise Applications and the Relational Model

The hierarchical data model used in early systems like IMS works well for simple transaction processing, but it has several shortcomings when it comes to analyzing the data stored in the database, a fundamental requirement for an enterprise application. In particular, combining information from different entity types can be inefficient if they are combined based on values stored in leaf nodes. In large hierarchies with many levels this takes a significant amount of time.

In 1970, Codd introduced a relational data model – and an associated relational calculus based on a set of algebraic operators – that was flexible enough to represent almost any dependencies and relations among entity types [8]. Codd’s model gained quick acceptance in academic circles, but it was not until 1974 – when a language initially called Structured English Query Language (SEQUEL) [9] was introduced, which allowed the formulation of queries against relational data in a comparatively user friendly language – that acceptance of the relational model became more widespread. Later, the description of the query language was shortened to Structured Query Language (SQL), and in conjunction with the relational model it could be used to serve a wide variety of applications. Systems based on this model became known as Relational Database Management Systems (RDBMS).

The next significant advance in RDBMS development came with the introduction of the concept of a transaction [10]. A transaction is a fixed sequence of actions with a well-defined beginning and a well-defined ending. This concept coined the term ACID [11], which describes the properties of a transaction. ACID is an acronym for the terms Atomicity, Consistency, Isolation, and Durability. Atomicity is the capability to make a set of different operations on the database appear as a single operation to the user, and all of the different operations should be executed or none at all. The consistency property ensures that the database is in a consistent state before the start and after the end of a transaction. In order to ensure consistency among concurrent transactions, isolation is needed, as it fulfills the requirement that only a transaction itself can access its intermediate data unless the transaction is not finished. Atomicity, consistency, and isolation affect the way data is processed by a DBMS. Durability guarantees that a successful transaction persists and is not affected by any kind of system failure. These are all essential features in an enterprise application.

RDBMSs supporting ACID transactions provided an efficient and reliable way for storing and retrieving enterprise data. Throughout the 1980s, customers found that they could use enterprise applications based on such systems to process their operational data, so-called Online Transaction Processing (OLTP), and for any analytics or Online Analytical Processing (OLAP), they needed [12]. In the rest of the book we will use the terms OLTP and the processing of operational data interchangeably, and the terms OLAP and analytical processing interchangeably.

Separation of Transaction and Analytical Processing

As data volumes grew, RDBMSs were no longer able to efficiently service the requirements of all categories of enterprise applications. In particular, it became impossible for the DBMS itself to service ad-hoc queries on the entire transactional database in a timely manner.

One of the reasons the DBMS was unable to handle these ad-hoc queries is the design of the database schemas that underlie most transactional enterprise applications. OLTP schemas are highly normalized to minimize the data entry volume and to speed up inserts, update and deletes. This high degree of normalization is a disadvantage when it comes to retrieving data, as multiple tables may have to be joined to get all the desired information. Creating these joins and reading from multiple tables can have a severe impact on performance, as multiple reads to disk may be required. Analytical queries need to access large portions of the whole database, which results in long run times with regard to traditional solutions.

OLAP systems were developed to address the requirement of large enterprises to analyze their data in a timely manner. These systems relied on specialized data structures [13] designed to optimize read performance and provide quick processing of complex analytical queries. Data must be transferred out of an enterprise's transactional system into an analytical system and then prepared for predefined reports.

The transfer happens in cyclic batches, in a so-called Extract, Transform, and Load (ETL) process [14]. The required reports may contain data from a number of different source systems. This must be extracted and converted into a single format that is appropriate for transformation processing. Rules are then applied during the transformation phase to make sure that the data can be loaded into the target OLAP system. These rules perform a number of different functions, for example, removing duplicates, sorting and aggregation. Finally, the transformed data is loaded into a target schema optimized for fast report generation.

This process has the severe limitation in that one is unable to do real-time analytics as the analytical queries are posed against a copy of the data in the OLAP system that does not include the latest transactions.

Performance Advantage of In-Memory Technology over Disk

The main reason that current RDBMSs cannot perform the required queries fast enough is that query data must be retrieved from disk. Modern systems make extensive use of caching to store frequently accessed data in main memory but for queries that process large amounts of data, disk reads are still required. Simply accessing and reading the data from disk can take a significant amount of time. Table 1.1 shows the access and read times for disk and main memory (based on [15]).

Table 1.1: Access and Read Times for Disk and Main Memory

Action	Time
Main Memory Access	100 ns
Read 1 MB Sequentially from Memory	250,000 ns
Disk Seek	5,000,000 ns
Read 1 MB Sequentially from Disk	30,000,000 ns

Main memory or in-memory databases have existed since the 1980s [16], but it is only recently that Dynamic Random Access Memory (DRAM) has become inexpensive enough to make these systems a viable option for large enterprise systems.

The ability of the database layer in an enterprise application to process large volumes of data quickly is fundamental to our aim of removing the need for a separate analytics systems. This will allow us to achieve our goal of providing a sub-second response time for any business query. In-memory databases based on the latest hardware can provide this functionality and they form the cornerstone of our proposed database architecture discussed in Chapter 3.

1.2.2 Main Memory Is the New Disk

Since in-memory databases utilize the server’s main memory as primary storage location, the size, cost, and access speed provided by main memory components are vitally important. With the help of data compression, today’s standard server systems comprise sufficiently large main memory volumes to accommodate the entire operational data of all companies (Section 4.3). Main memory as the primary storage location is becoming increasingly attractive as a result of the decreasing cost/size ratio. The database can be directly optimized for main memory access, omitting the implementation of special algorithms to optimize disk access.

Figure 1.2 provides an overview of the development of main memory, disk, and flash storage prices over time. The cost/size relation for disks as well as main memory has decreased exponentially in the past. For example, the price for 1 MB of disk space dropped below US \$ 0.01 in 2001, which is a rapid decrease compared to the cost of more than US \$ 250 in 1970. A similar development can be observed for main memory. In addition to the attractiveness of fitting all operational business data of a company into main memory, optimizing and simplifying data access accordingly, the access speed of main memory compared to that of disks is four orders of magnitude faster: A main memory reference takes 100 ns [17]. Current disks typically provide read and write seek times of about 5 ms [18, 19].

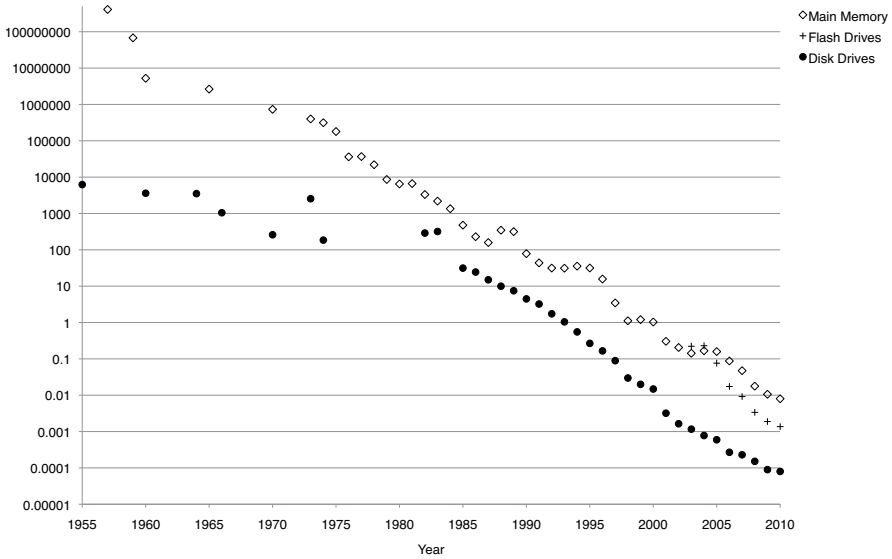


Figure 1.2: Storage Price Development

1.2.3 From Maximizing CPU Speed to Multi-Core Processors

In 1965, Intel co-founder Gordon E. Moore made his famous prediction about the increasing complexity of integrated circuits in the semiconductor industry [20]. The prediction became known as Moore's Law, and has become shorthand for rapid technological change. Moore's Law states that the number of transistors on a single chip is doubled approximately every two years [21].

In reality, the performance of Central Processing Units (CPUs) doubles every 20 months on average. The brilliant achievement that computer architects have managed is not only creating faster transistors, which results in increased clock speeds, but also in an increased number of transistors per CPU per square meter, which became cheaper due to efficient production methods and decreased material consumption. This leads to higher performance for roughly the same manufacturing cost. For example, in 1971, a processor consisted of 2300 transistors whereas in 2006 it consisted of about 1.7 billion transistors at approximately the same price. Not only does an increased number of transistors play a role in performance gain, but also more efficient circuitry. A performance gain of up to a factor of two per core has been reached from one generation to the next, while the number of transistors remained constant.

Figure 1.3 provides an overview of the development of processor clock speed and the number of transistors from 1971 to 2010 based on [22, 23, 24]. As shown, the clock speed of processors had been growing exponentially for almost 30 years, but has stagnated since 2002. Power consumption, heat distribution and dissipation, and the speed of light have become the limiting factors for Moore's Law [25].

The Front Side Bus (FSB) speed, having grown exponentially in the past, has also stagnated.

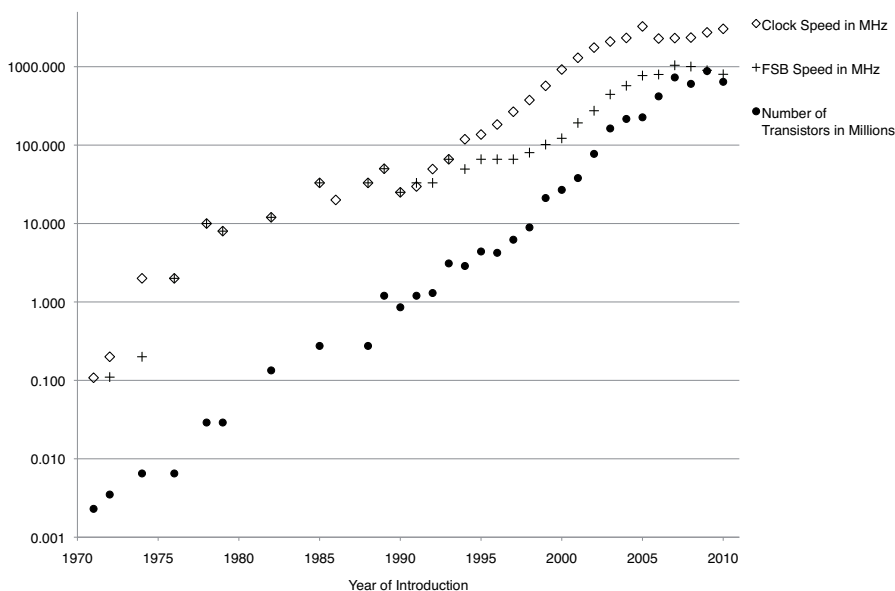


Figure 1.3: Clock Speed, FSB Speed, and Transistor Development

In 2001, IBM introduced the first processor on one chip, which was able to compute multiple threads at the same time independently. The IBM Power 4 [26] was built for the high-end server market and was part of IBM's Regatta Servers. Regatta was the code-name for a module containing multiple chips, resulting in eight cores per module [27]. In 2002, Intel introduced its proprietary hyper-threading technology, which optimizes processor utilization by providing thread-level parallelism on a single core. With hyper-threading technology, multiple logical processors with duplicated architectural state are created from a single physical processor. Several tasks can be executed virtually in parallel, thereby increasing processor utilization. Yet, the tasks are not truly executed in parallel because the execution resources are still shared and only multiple instructions of different tasks that are compatible regarding resource usage can be executed in a single processing step. Hyper-threading is applicable to single-core as well as to multi-core processors.

Until 2005, single-core processors dominated the home and business computer domain. For the consumer market, multi-core processors were introduced in 2005 starting with two cores on one chip, for example, Advanced Micro Devices' (AMD) Athlon 64 X2. An insight into the development of multi-core processors and future estimates of hardware vendors regarding the development of multi-core technology is provided in Figure 1.4. At its developer forum in autumn 2006, Intel presented a prototype for an 80-core processor, while IBM introduced the Cell Broadband Engine with ten cores in the same year [28]. The IBM Cell Broadband Engine consists of two

different types of processing elements, one two-core PowerPC processing element and up to eight synergistic processing elements that aim at providing parallelism at all abstraction levels. In 2008, Tiler introduced its Tile64, a multi-core processor for the high-end embedded systems market that consists of 64 cores [29]. 3Leaf is offering a product that is based on the HyperTransport architecture [30] with 192 cores. In the future, higher numbers of cores are anticipated on a single chip. In 2008, Tiler predicted a chip with 4096 cores by 2017 for the embedded systems market and Sun estimated that servers are going to feature 32 and up to 128 cores by 2018 [31].

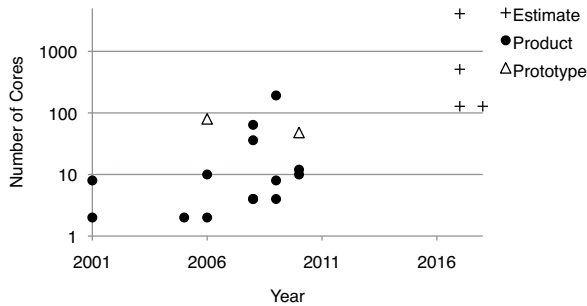


Figure 1.4: Development of Number of Cores

The developers of current and future software systems have to take multi-core and multi-processor machines into account. Software programs can no longer implicitly utilize the advances in processor technology, as was the case in the past decades with the growth of processor clock speed. In other words, “the free lunch is over” [32]. Utilization of parallelism through many processing units has to be explicitly incorporated into software by either splitting up algorithms across several computing units or executing many operations concurrently, even on the same set of data in case of a database system (Section 4.2). A programming paradigm, which scales well with an increasing number of single processing units, includes lock-free data structures and algorithms [33] (Section 5.5). With the development of multi-core architectures the stagnation of clock speed per core is compensated and technology advances as well as software development are heading in a new direction.

1.2.4 Increased Bandwidth between CPU and Main Memory

The increased performance of the FSB, which so far has been the only interface from the CPU to main memory and all other input/output (I/O) components, no longer keeps up with the exponential growth of processor performance anymore, as can be seen in Figure 1.3. Important for the calculation of the theoretical memory throughput are clock speed (cycles per second) and bandwidth of the data bus.

The increased clock speed and the use of multiple cores per machine are resulting in a widening gap between the ability of processors to digest data and the ability of the infrastructure to provide data. In-memory and column-oriented

data storage enable the usage of additional processing power despite the bottleneck created by the aforementioned widening gap. High compression rates of column-oriented storage can lead to a better utilization of bandwidth. In-memory data storage can utilize enhanced algorithms for data access, for example, prefetching. We will discuss in-memory and column-oriented storage for database systems later in this book (Chapter 4).

Using compressed data and algorithms that work on compressed data is standard technology and has already proven to be sufficient to compensate the data supply bottleneck for machines with a small number of cores. It is, however, failing with the addition of many more cores. Experiments with column-oriented, compressed in-memory storage and data-intensive applications showed that the FSB was well utilized, though not yet congested, in an eight-core machine. The data processing requirements of the same applications on a 24-core machine surmounted the FSB's ability to provide enough data. From these experiments we can see that new memory access strategies are needed for machines with even more cores to circumvent the data supply bottleneck. Processing resources are often underutilized and the growing performance gap between memory latency and processor frequency intensifies the underutilization [34].

Figure 1.3 provides an overview of the development of the FSB speed. Intel improved the available transfer rate, doubling the amount of data that can be transferred in one cycle or added additional independent buses on multi-processor boards. The HyperTransport protocol was introduced by AMD in 2001 [30] to integrate the memory controller into the processor. Similar to the HyperTransport protocol, Intel introduced Quick Path Interconnect (QPI) [35] in the second half of 2008. QPI is a point-to-point system interconnect interface for memory and multiple processing cores, which replaces the FSB. Every processor has one or multiple memory controllers with several channels to access main memory in addition to a special bus to transfer data among processors. Compared to Intel FSB in 2007 with a bandwidth of 12.8 GB per second, QPI helped to increase the available bandwidth to 25.6 GB per second in 2008 [35]. In Intel's Nehalem EP chips, each processor has three channels from the memory controller to the physical memory [36]. In Intel's Nehalem EX chips, these channels have been expanded to four channels per processor [37].

Figure 1.5 gives an overview of the different architectures. In QPI, as shown in Figure 1.5 (b), every processor has its exclusively assigned memory. On an Intel XEON 7560 (Nehalem EX) system with four processors, benchmark results have shown that a throughput of more than 72 GB per second is possible [37]. In contrast to using the FSB, shown in Figure 1.5 (a), the memory access time differs between local memory (adjacent slots) and remote memory that is adjacent to the other processing units. As a result of this characteristic, architectures based on the FSB are called Uniform Memory Access (UMA) and the new architectures are called Non-Uniform Memory Access (NUMA). We differentiate between cache-coherent NUMA (ccNUMA) and non cache-coherent NUMA systems. In ccNUMA systems, all CPU caches share the same view to the available memory and coherency is ensured by a protocol implemented in hardware. Non cache-coherent NUMA

systems require software layers to take care of conflicting memory accesses. Since most of the available standard hardware only provides ccNUMA, we will solely concentrate on this form.

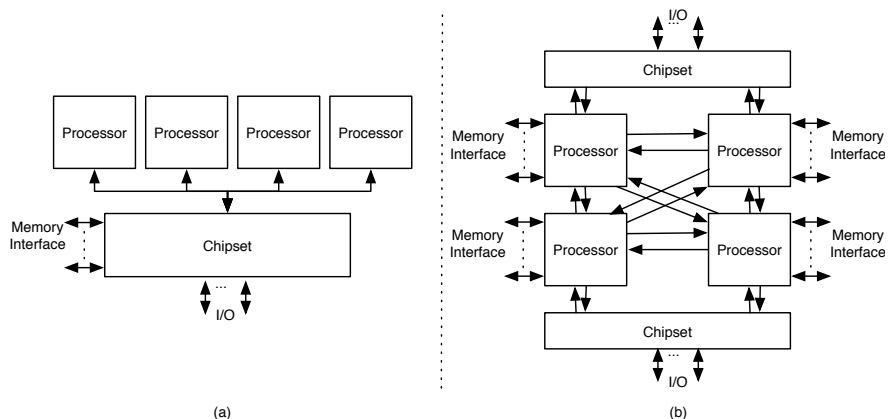


Figure 1.5: (a) Shared FSB, (b) Intel Quick Path Interconnect [35]

To exploit NUMA completely, applications have to be made aware of primarily loading data from the locally attached memory slots of a processor. Memory-bound applications might see a degradation of up to 25% of their performance if only remote memory is accessed instead of the local memory [37]. Reasons for this degradation can be the saturation of the QPI link between processor cores to transport data from the adjacent memory slot of another core, or the influence of higher latency of a single access to a remote memory slot. The full degradation might not be experienced, as memory caches and prefetching of data mitigates the effects of local versus remote memory. Assume a job can be split into many parallel tasks. For the parallel execution of these tasks distribution of data is relevant. Optimal performance can only be reached if the executed tasks solely access local memory. If data is badly distributed and many tasks need to access remote memory, the connections between the processors can become flooded with extensive data transfer.

Aside from the use for data-intensive applications, some vendors use NUMA to create alternatives for distributed systems. Through NUMA, multiple physical machines can be consolidated into one virtual machine. Note the difference in the commonly used term of virtual machine, where part of a physical machine is provided as a virtual machine. With NUMA, several physical machines fully contribute to the one virtual machine giving the user the impression of working with an extensively large server. With such a virtual machine, the main memory of all nodes and all CPUs can be accessed as local resources. Extensions to the operating system enable the system to efficiently scale out without any need for special remote communication that would have to be handled in the operating system or the applications. In most cases, the remote memory access is improved

by the reservation of some local memory to cache portions of the remote memory. Further research will show if these solutions can outperform hand-made distributed solutions. 3Leaf, for example, is a vendor that uses specialized hardware. Other companies, for example, ScaleMP [38] rely on pure software solutions to build virtual systems. In summary, we have observed that the enhancement of the clock speed of CPU cores has tended to stagnate, while adding more cores per machine is now the reason for progress. As we have seen, increasing the number of cores does not entirely solve all existing problems, as other bottlenecks exist, for example, the gap between memory access speed and the clock speed of CPUs. Compression reduces the effects of this gap at the expense of computing cycles. NUMA as an alternative interconnection strategy for memory access through multiple cores has been developed. Increased performance through the addition of more cores and NUMA can only be utilized by adapting the software accordingly. In-memory databases in combination with column-oriented data storage are particularly well suited for multi-core architectures. Column-oriented data storage inherently provides vertical partitioning that supports operator parallelism (Section 4.2).

1.3 Reducing Cost through In-Memory Data Management

In this section, we discuss the viability of using in-memory technology for enterprise data management and the financial costs of setting up and running an enterprise system. After an overview of the major cost factors, we look at how an architecture based on in-memory technology can help to reduce costs.

1.3.1 Total Cost of Ownership

The Total Cost of Ownership (TCO) is a business formula designed to estimate the lifetime costs of acquiring and operating resources, which in our case is an enterprise software system. The decision as to which hardware or software will be acquired and implemented will have a serious impact on the business. It is crucial to obtain an accurate cost estimate. The TCO measure was introduced when it became obvious that it is not sufficient to base IT decisions solely on the acquisition costs of the equipment because a substantial part of the cost is incurred later in the system lifecycle. The TCO analysis includes direct costs, for example, hardware acquisition, and indirect costs such as training activities for end users [39].

The primary purpose of introducing TCO is to identify all hidden cost factors and to supply an accurate and transparent cost model. This model can help to identify potential cost problems early. It is also a good starting point for a return on investment calculation or a cost-benefit analysis. These business formulas go beyond the TCO analysis and take the monetary profit or other benefits into consideration. Both are used to support decisions about technology changes or optimization activities.

Estimating TCO is a challenging exercise. Particular difficulties lie in creating an accurate cost model and estimating hidden costs. In many cases, the TCO

analysis leads to a decision between a one-time investment on the one hand and higher ongoing costs on the other hand. This is known as the TCO tradeoff [40]. An investment in centralization and standardization helps to simplify operations and reduces overhead, thus reducing the cost of support, upgrades, and training. As another example, an initial investment in expensive high-end hardware can boost system performance and facilitate development, administration, and all other operations.

1.3.2 Cost Factors in Enterprise Systems

The cost factors in setting up and maintaining an enterprise system include the cost of buying and operating the hardware infrastructure, as well as the cost of buying, administrating, and running the software.

Hardware Infrastructure and Power Consumption

The kind of hardware needed depends on the specific requirements of the applications. Enterprise applications of a certain complexity require high availability and low response time for many users on computationally complicated queries over huge amounts of data. The occurrence of many challenging requirements typically implies the need for high-end hardware.

Power costs for servers and for cooling systems constitute a major part of the ongoing costs. The cost for power and the infrastructure needed for power distribution and cooling makes up about 30% of the total monthly cost of running a large data center, while the server costs, which are amortized over a shorter time span of three years, constitute close to 60%. These numbers refer to very large data centers (about 50,000 servers) and if their power usage efficiency is very high [41]. So, in many smaller data centers, the power and power infrastructure might contribute a greater portion of the cost.

System Administration

The cost impact of system administration is largely determined by the time and human effort it requires. These factors are determined by the performance of the system and the complexity of the tasks. The first tasks include the initial system setup and the configuration and customization of the system according to the business structure of the customer. Upgrades and extensions can affect every single component in the system, so it is a direct consequence that they get more expensive as the system gets more complex. The same is true for the monitoring of all system components in order to discover problems (like resource bottlenecks) early. The costs and the risks involved in scaling and changing a complex system are limiting factors for upgrading or extending an existing system.

Typical administrative tasks involve scanning, transforming, or copying large amounts of data. As an example, when a new enterprise system is set up, it is often necessary to migrate data from a legacy system. During system upgrades and

extensions, complete tables might need to be transformed and reformatted. For system backups as well as for testing purposes complete system copies are created.

1.3.3 In-Memory Performance Boosts Cost Reduction

If all data can be stored in main memory instead of on disk, the performance of operations on data, especially on mass data, is improved. This has impact on every aspect of the system: it affects the choices of hardware components, but also the software architecture and the basic software development paradigms.

Many performance crutches, like redundant materialized data views, have been introduced solely to optimize response time for analytical queries. The downside is that redundancy has to be maintained with significant overhead (Section 7.1). This overhead becomes insignificant when the system is fast enough to handle requests to scan huge amounts of data on the fly. Pushing data-intensive operations into the database simplifies the application software stack by fully utilizing the functionality of the database and by avoiding the necessity of transporting massive amounts of data out of the database for computation. If data migration and operations on mass data are accelerated, this automatically makes system copying, backups, archiving, and upgrade tasks less time consuming, thus reducing cost.

The improved performance of operations on mass data not only facilitates analytical queries, but also many of the day-to-day administrative tasks mentioned above. We have indicated that a simplification of a software system's architecture has multiple positive effects on the cost. In short, a simpler system, for example, an architecture with fewer layers and fewer components, is less expensive and faster to set up, easier to operate, easier to scale and change, and it generates fewer failures.

Our current experience in application development on the basis of in-memory technology shows that the size of the application code can be reduced up to 75%. All the orchestration of layers, for example, caches and materialization, is not required any longer and algorithms are pushed down to the database to operate close to the data needed.

With respect to user-system interaction, improved performance and reduced complexity directly translate into reduced cost. The increased cost through wasted time that occurs whenever someone has to wait for a response from the system or has to invest time to understand a complex process affects all stages in the software lifecycle, from software development and system administration to the end user of the business application.

Compared to high-end disk-based systems that provide the necessary performance for database computing, the initial costs of in-memory systems, as well as running costs, are not that different. High-end disk-based systems provide the necessary bandwidth for computation through redundancy of disks. Although disks are able to provide more space less expensively, scaling along the number of disks and managing them is mandatory for comparable performance to in-memory computing.

In summary, we can say that using in-memory technology can help to reduce the TCO of an enterprise application by reducing the complexity of the application

software layers, performing data-intensive tasks close to the data source, speeding up response times allowing users to make more efficient use of their time, as well as speeding up administrative tasks involving mass data.

1.4 Conclusion

In-memory and multi-core technology have the potential to improve the performance of enterprise applications and the value they can add to a business. In this chapter we described the potential impact of in-memory technology on enterprise applications and why it is the right time to make the switch.

We first identified desirable new features that can be added to enterprise applications if performance is improved. Chief among them was the ability to perform analytics on transactional data rather than having to use a separate BI system for analysis.

Having identified these features, we then looked at the feasibility of providing them in an enterprise application by using a redesigned DBMS based on in-memory technology. We showed how current trends in hardware can be utilized to realize our vision and how these new hardware developments complement our database design.

Finally, we discussed the viability of our approach by assessing the impact that using a DBMS based on in-memory technology has on the TCO of an enterprise system. We ended by describing how the performance increase made possible by such a system could reduce the TCO.