Practical Hadoop Migration

How to Integrate Your RDBMS with the Hadoop Ecosystem and Re-Architect Relational Applications to NoSQL

Bhushan Lakhe
To my mother....
# Contents at a Glance

- Foreword .................................................................................................................. xv
- About the Author ........................................................................................................ xvii
- About the Technical Reviewer ................................................................................... xix
- Acknowledgments ........................................................................................................ xxi
- Introduction ................................................................................................................... xxiii

- Chapter 1: RDBMS Meets Hadoop: Integrating, Re-Architecting, and Transitioning .......................................................... 1

  - Chapter 2: Understanding RDBMS Design Principles ........................................... 27
  - Chapter 3: Using SSADM for Relational Design .................................................... 53
  - Chapter 4: RDBMS Design and Implementation Tools .......................................... 89

- Part II: Hadoop: A Review of the Hadoop Ecosystem, NoSQL Design Principles and Best Practices .................................................. 101
  - Chapter 5: The Hadoop Ecosystem ........................................................................ 103
  - Chapter 6: Re-Architecting for NoSQL: Design Principles, Models and Best Practices .......................................................... 117
<table>
<thead>
<tr>
<th>CONTENTS AT A GLANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ Part III: Integrating Relational Database Management Systems with the Hadoop Distributed File System..... 149</td>
</tr>
<tr>
<td>■ Chapter 7: Data Lake Integration Design Principles......................... 151</td>
</tr>
<tr>
<td>■ Chapter 8: Implementing SQOOP and Flume-based Data Transfers.......................................................... 189</td>
</tr>
<tr>
<td>■ Part IV: Transitioning from Relational to NoSQL Design Models ........................................................... 207</td>
</tr>
<tr>
<td>■ Chapter 9: Lambda Architecture for Real-time Hadoop Applications.......................................................... 209</td>
</tr>
<tr>
<td>■ Chapter 10: Implementing and Optimizing the Transition............ 253</td>
</tr>
<tr>
<td>■ Part V: Case Study for Designing and Implementing a Hadoop-based Solution ........................................ 277</td>
</tr>
<tr>
<td>■ Chapter 11: Case Study: Implementing Lambda Architecture.... 279</td>
</tr>
<tr>
<td>Index.................................................................................................................. 303</td>
</tr>
</tbody>
</table>
Contents

Foreword ......................................................................................... xv
About the Author ........................................................................... xvii
About the Technical Reviewer ........................................................ xix
Acknowledgments .......................................................................... xxi
Introduction .................................................................................. xxiii

Chapter 1: RDBMS Meets Hadoop: Integrating, Re-Architecting, and Transitioning 1

Conceptual Differences Between Relational and HDFS NoSQL Databases 2

Relational Design and Hadoop in Conjunction:
Advantages and Challenges ......................................................... 6
  Type of Data ................................................................................... 9
  Data Volume .................................................................................. 9
  Business Need ............................................................................. 10

Deciding to Integrate, Re-Architect, or Transition 10
  Type of Data .................................................................................. 10
  Type of Application ..................................................................... 11
  Business Objectives ................................................................... 12

How to Integrate, Re-Architect, or Transition 13
  Integration .................................................................................... 13
  Re-Architecting Using Lambda Architecture ............................. 16
  Transition to Hadoop/NoSQL ..................................................... 21

Summary ..................................................................................... 23

Chapter 2: Understanding RDBMS Design Principles ...................... 27

Overview of Design Methodologies ................................................................. 28
  Top-down ....................................................................................................... 28
  Bottom-up ..................................................................................................... 29
  SSADM ......................................................................................................... 29

Exploring Design Methodologies ............................................................. 30
  Top-down ....................................................................................................... 30
  Bottom-up ..................................................................................................... 34
  SSADM ......................................................................................................... 36

Components of Database Design .............................................................. 40
  Normal Forms .............................................................................................. 41
  Keys in Relational Design ........................................................................... 45
  Optionality and Cardinality ....................................................................... 46
  Supertypes and Subtypes ........................................................................... 48

Summary ...................................................................................................... 51

Chapter 3: Using SSADM for Relational Design ............................. 53

Feasibility Study ............................................................................................ 54
  Project Initiation Plan .................................................................................. 55
  Requirements and User Catalogue .............................................................. 58
  Current Environment Description .............................................................. 61
  Proposed Environment Description ............................................................ 63
  Problem Definition ....................................................................................... 65
  Feasibility Study Report ............................................................................. 66
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Analysis</td>
<td>68</td>
</tr>
<tr>
<td>Investigation of Current Environment</td>
<td>68</td>
</tr>
<tr>
<td>Business System Options</td>
<td>74</td>
</tr>
<tr>
<td>Requirements Specification</td>
<td>75</td>
</tr>
<tr>
<td>Data Flow Model</td>
<td>75</td>
</tr>
<tr>
<td>Logical Data Model</td>
<td>77</td>
</tr>
<tr>
<td>Function Definitions</td>
<td>78</td>
</tr>
<tr>
<td>Effect Correspondence Diagrams (ECDs)</td>
<td>79</td>
</tr>
<tr>
<td>Entity Life Histories (ELHs)</td>
<td>81</td>
</tr>
<tr>
<td>Logical System Specification</td>
<td>83</td>
</tr>
<tr>
<td>Technical Systems Options</td>
<td>83</td>
</tr>
<tr>
<td>Logical Design</td>
<td>84</td>
</tr>
<tr>
<td>Physical Design</td>
<td>86</td>
</tr>
<tr>
<td>Logical to Physical Transformation</td>
<td>86</td>
</tr>
<tr>
<td>Space Estimation Growth Provisioning</td>
<td>87</td>
</tr>
<tr>
<td>Optimizing Physical Design</td>
<td>87</td>
</tr>
<tr>
<td>Summary</td>
<td>88</td>
</tr>
<tr>
<td>Chapter 4: RDBMS Design and Implementation Tools</td>
<td>89</td>
</tr>
<tr>
<td>Database Design Tools</td>
<td>90</td>
</tr>
<tr>
<td>CASE tools</td>
<td>90</td>
</tr>
<tr>
<td>Diagramming Tools</td>
<td>95</td>
</tr>
<tr>
<td>Administration and Monitoring Applications</td>
<td>96</td>
</tr>
<tr>
<td>Database Administration or Management Applications</td>
<td>97</td>
</tr>
<tr>
<td>Monitoring Applications</td>
<td>98</td>
</tr>
<tr>
<td>Summary</td>
<td>99</td>
</tr>
</tbody>
</table>
Part III: Integrating Relational Database Management Systems with the Hadoop Distributed File System

Chapter 7: Data Lake Integration Design Principles

Data Lake vs. Data Warehouse

Data Warehouse

Data Lake

Concept of a Data Lake

Data Reservoirs

Exploratory Lakes

Analytical Lakes

Factors for a Successful Implementation

Summary

Chapter 8: Implementing SQOOP and Flume-based Data Transfers

Deciding on an ETL Tool

Sqoop vs. Flume

Processing Streaming Data

Using SQOOP for Data Transfer

Using Flume for Data Transfer

Flume Architecture

Understanding and Using Flume Components

Implementing Log Consolidation Using Flume

Summary
Part IV: Transitioning from Relational to NoSQL Design Models ............................................................ 207

Chapter 9: Lambda Architecture for Real-time Hadoop Applications ........................................................................ 209

Defining and Using the Lambda Layers ............................................................... 210
Batch Layer ........................................................................................................ 211
Serving Layer ................................................................................................... 224
Speed Layer ................................................................................................. 229

Pros and Cons of Using Lambda ................................................................. 234
Benefits of Lambda ...................................................................................... 234
Issues with Lambda ..................................................................................... 235
The Kappa Architecture .............................................................................. 236

Future Architectures1 ..................................................................................... 238
A Bit of History ............................................................................................ 238
Butterfly Architecture ................................................................................ 240

Summary ................................................................................................. 250

Chapter 10: Implementing and Optimizing the Transition ................. 253
Hardware Configuration ............................................................................... 254
Cluster Configuration .................................................................................. 254

Operating System Configuration ................................................................. 255

Hadoop Configuration .................................................................................. 257
HDFS Configuration .................................................................................... 258
Choosing an Optimal File Format ................................................................. 266
Indexing Considerations for Performance ...................................................... 274

Choosing a NoSQL Solution and Optimizing Your Data Model ............. 275

Summary ................................................................................................. 276
Part V: Case Study for Designing and Implementing a Hadoop-based Solution .............................................. 277

Chapter 11: Case Study: Implementing Lambda Architecture.... 279

The Business Problem and Solution.............................................................. 280

Solution Design .......................................................................................... 280

Hardware ...................................................................................................... 280

Software ...................................................................................................... 282

Database Design .......................................................................................... 282

Implementing Batch Layer ......................................................................... 286

Implementing the Serving Layer ................................................................. 289

Implementing the Speed Layer .................................................................... 292

Storage Structures (for Master Data and Views) ......................................... 296

Other Performance Considerations ............................................................. 297

Reference Architectures ............................................................................... 298

Changes to Implementation for Latest Architectures ................................. 299

Summary ...................................................................................................... 301

Index .......................................................................................................... 303
We are in the midst of one of the biggest transformations of Information Technology (IT). Rapidly evolving business requirements have demanded agility in all aspects of IT. As more and more paper-based business processes are getting digital, rapid application development, staging, and deployment have become the norm. In addition, the data exhaust from these digital applications has become enormous and needs to be analyzed in real time. Growing volumes of historical data is considered valuable for improving business efficiency and identifying future trends and disruptions. Ubiquitous end-user connectivity, cost-efficient software and hardware sensors, and democratization of content production have led to the deluge of data generated in enterprises. As a result, the traditional data infrastructure has to be revamped. Of course, this cannot be done overnight. To prepare your IT to meet the new requirements of the business, one has to carefully plan re-architecting the data infrastructure so that existing business processes remain available during this transition.

Hadoop and NoSQL platforms have emerged in the last decade to address the business requirements of large web-scale companies. Capabilities of these platforms are evolving rapidly, and, as a result, have created a lot of hype in the industry. However, none of these platforms is a panacea for all the needs of a modern business. One needs to carefully consider various business use cases and determine which platform is most suitable for each specific use case. Introducing immature platforms for use cases that are not suited for them is the leading cause of failure of data infrastructure projects. Data architects of today need to understand a variety of data platforms, their design goals, their current and future data protection capabilities, access methods, and performance sweet spots, and how they compare in features against traditional data platforms. As a result, traditional database administrators and business analysts are overwhelmed by the sheer number of new technologies and the rapidly changing data landscape.

This book is written with those readers in mind. It cuts through the hype and gives a practical way to transition to the modern data architectures. Although it may feel like new technologies are emerging every day, the key to evaluating these technologies is to align your current and future business use cases and requirements to the design-center of these new technologies. This book helps readers understand various aspects of the modern data platforms and helps navigate the emerging data architecture. I am confident that it will help you avoid the complexity of implementing modern data architecture and allow seamless transition for your business.

—Milind Bhandarkar, PhD
Founder and CEO, Ampool, Inc.
Milind Bhandarkar was the founding member of the team at Yahoo! that took Apache Hadoop from 20-node prototype to datacenter-scale production system, and has been contributing and working with Hadoop since version 0.1.0. He started the Yahoo! Grid solutions team focused on training, consulting, and supporting hundreds of new migrants to Hadoop. Parallel programming languages and paradigms has been his area of focus for over 20 years. He has worked at the Center for Development of Advanced Computing (C-DAC), National Center for Supercomputing Applications (NCSA), Center for Simulation of Advanced Rockets, Siebel Systems, Pathscale Inc. (acquired by QLogic), Yahoo!, and Linkedin. Until 2013, Milind was chief architect at Greenplum Labs, a division of EMC. Most recently, he was chief scientist at Pivotal Software. Milind holds his PhD degree in computer science from the University of Illinois at Urbana-Champaign.
About the Author

Bhushan Lakhe is a Big Data professional, technology evangelist, author, and avid blogger who resides in the windy city of Chicago. After graduating in 1988 from one of India’s leading universities (Birla Institute of Technology and Science, Pilani), he started his career with India’s biggest software house, Tata Consultancy Services. Thereafter, he joined ICL, a British computer company, and worked with prestigious British clients. Moving to Chicago in 1995, he worked as a consultant with Fortune 50 companies like Leo Burnett, Blue Cross, Motorola, JPMorgan Chase, and British Petroleum, often in a critical and pioneering role.

After a seven-year stint executing successful Big Data (as well as data warehouse) projects for IBM’s clients (and receiving the company’s prestigious Gerstner Award in 2012), Mr. Lakhe spent two years helping Unisys Corporation’s clients with Big Data implementations, and thereafter two years as senior vice president (information and data architecture) at Ipsos (the world’s third-largest market research corporation), helping design global data architecture and Big Data strategy.

Currently, Mr. Lakhe heads the Big Data practice for HCL America, a $7 billion global consulting company with offices in 31 countries. At HCL, Mr. Lakhe is involved in architecting Big Data solutions for Fortune 500 corporations. Mr. Lakhe is active in the Chicago Hadoop community and is co-organizer for a Meetup group (www.meetup.com/ambariCloud-Big-Data-Meetup/) where he regularly talks about new Hadoop technologies and tools. You can find Mr. Lakhe on LinkedIn at www.linkedin.com/in/bhushanlakhe.
Robert L. Geiger is currently Chief Architect and acting VP of engineering at Ampool Inc., an early stage startup in the Big Data and analytics infrastructure space. Before joining Ampool, he worked as an architect and developer in the solutions/SaaS space at a B2B deep learning based startup, and prior to that as an architect and team lead at Pivotal Inc., working in the areas of security and analytics as a service for the Hadoop ecosystem. Prior to Pivotal, Robert served as a developer and VP, engineering at a small distributed database startup, TransLattice. Robert spent several years in the security space working on and leading teams in at Symantec on distributed intrusion detection systems. His career started with Motorola Labs in Illinois where he worked on distributed IP over wireless systems, crypto/security, and e-commerce after graduating from University of Illinois Champaign-Urbana.
Acknowledgments

This is my second book for Apress (the first being *Practical Hadoop Security*) continuing the *Practical Hadoop* series, and I want to thank Apress for giving me the opportunity to write it. I would like to thank the Hadoop community and the user forums that bring innovation to this technology and keep the world interested! I have learned a lot from the selfless people in the Hadoop community who believe in being Good Samaritans.

On a personal note, I want to thank my friend Satya Kondapalli for making a forum of Hadoop enthusiasts available through our Meetup group Ambaricloud. I also want to thank our sponsors Hortonworks for supporting us. Finally, I would like to thank my friend Milind Bhandarkar (of Ampool) for taking time from his busy schedule to write a foreword and a whole section about his new Butterfly architecture.

I am grateful to my editors, Rita Fernando, Robert Hutchinson, and Matthew Moodie at Apress for their help in getting this book together. Rita has been there throughout to answer any questions that I have, to improve my drafts, and to keep me on schedule. Robert Hutchinson’s help with the book structure has been immensely valuable. And I am also very thankful to Robert Geiger for taking time to review my second book technically. Bob always had great suggestions for improving a topic, recommending additional details, and of course resolving technical shortcomings.

Finally, the writing of this book wouldn’t have been possible without the constant support from my family (my wife, Swati, and my kids, Anish and Riya) for the second time in the last three years, and I’m looking forward to spending lots more time with all of them.
I have spent more than 20 years consulting for large corporations, and when I started, it was just relational databases. Eventually, the volumes of accumulated historical data grew, and it was not possible to manage and analyze this data with good performance. So, corporations started thinking about separating the parts (of data) useful for analysis (or generating insights) from the descriptive data. They soon realized that a fundamental change was needed in the relational design, and a new paradigm called data warehousing was born. Thanks to the work done by Bill Inmon and Ralph Kimball, the world started thinking (and designing) in terms of Star schemas and dimensions and facts. ETL (extract, transform, load) processes were designed to load the data warehouses.

The next step was making sure that large volumes of data could be retrieved with good performance. Specialized software was developed, and RDBMS solutions (Oracle, Sybase, SQL Server) added processing for data warehouses. For the next level of performance, it was clear that data needed to be preprocessed, and data cubes were designed. Since magnetic disk drives were slow, SSDs (solid state devices) were designed, and software that cached (or held data in RAM) data for speed of processing and retrieval became popular. So, with all these advanced measures for performance, why is Hadoop or NoSQL needed? For two reasons.

First, it is important to note that all this while, the data being processed either was relational data (for RDBMS) or had started as relational data (for data warehouses). This was structured data, and the type of analysis (and insights) possible was very specific (to the application that generated the data). The rigid structure of a warehouse put severe limits on the insights or data explorations that were possible, since you start with a design and fit data into it. Also, due to the very high volumes, warehouses couldn’t perform per expectations, and a newer technology was needed to effectively manage this data.

Second, in recent years, new types of data were introduced: unstructured or semi-structured data. Social media became very popular and were a new avenue for corporations to communicate directly with people once they realized the power behind it. Corporations wanted to know what people thought about their products, services, employees, and of course the corporations themselves. Also, with e-commerce forming a large part of all the businesses, corporations wanted to make sure they were preferred over their competitors—and if that was not the case, they wanted to know why. Finally, there was a need to analyze some other types of unstructured data, like sensor data from electrical and electronic devices, or data from mobile devices sensors, that was also very high volume. All this data was usually hundreds of gigabytes per day.

Conventional warehouse technology was incapable of processing or managing this data. So, a new technology had to be designed to process it, and with good performance (since total volumes were in terabytes). In some cases, as the unstructured data (or insights from it) needed to be combined with structured data, the new technology needed to support interfacing with data warehouses or RDBMS.
Hadoop offers all these capabilities and in addition allows a schema-on-read (meaning you can define metadata while performing analysis) that offers a lot of flexibility for performing exploratory analysis or generating new insights from your data.

This gets us to the final question: how do you migrate or integrate your existing RDBMS-based applications with Hadoop and analyze structured as well as unstructured data in tandem? Well, you have to read rest of the book to know that!

**Who This Book Is For**

This book is an excellent resource for IT management planning to migrate or integrate their existing RDBMS environment with Big Data technologies or Big Data architects who are designing a migration/integration process. This book is also for Hadoop developers who want to implement migration/integration process or students who'd like to learn about designing Hadoop applications that can successfully process relational data along with unstructured data. This book assumes a basic understanding of Hadoop, Kerberos, relational databases, Hive, Spark, and an intermediate level understanding of Linux.

**Downloading the Code**

The source code for this book is available in ZIP file format in the Downloads section of the Apress Web site (www.apress.com/9781484212882).

**Contacting the Author**

You can reach Bhushan Lakhe at blakhe@aol.com or bclakhe@gmail.com.