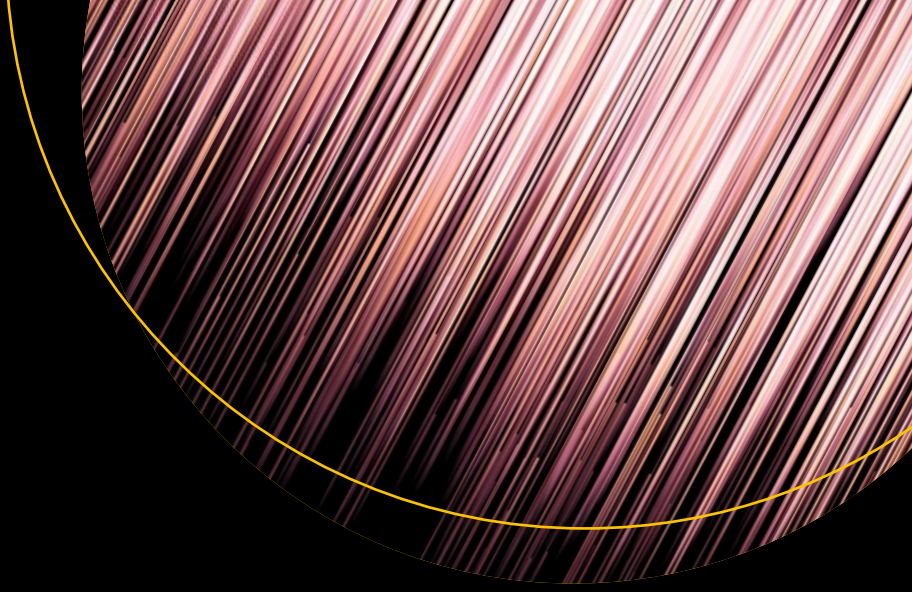

Migrating to MariaDB

Toward an Open Source Database Solution

—

William Wood

Apress®

Migrating to MariaDB

Toward an Open Source Database Solution

William Wood

Apress®

Migrating to MariaDB: Toward an Open Source Database Solution

William Wood
Pacific, MO, USA

ISBN-13 (pbk): 978-1-4842-3996-4 ISBN-13 (electronic): 978-1-4842-3997-1
https://doi.org/10.1007/978-1-4842-3997-1

Library of Congress Control Number: 2018963756

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484239964. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

First and foremost, I would like to dedicate this book to my son, Cam Carter, in that over many months there were times daddy could not do things that a four-year-old really needs to do. Someday he will understand, but for now I have lots of things to catch up on.

Next would be to my wonderful wife and her patience when I had to skip out on some family function, or something or another, in an attempt to get words on paper.

Finally, my boss, Phil Mazza, for providing me the opportunity to take on some interesting tasks and responsibilities. The fine folks at MariaDB for their database solution and allowing me to speak at their 2018 conference, and to the team at Apress for their patience and taking a chance on someone who has never done anything like this before. This dedication would of course be remiss without mentioning that none of this would be possible without the drivers for change.

Table of Contents

About the Author ix

About the Technical Reviewer xi

Introduction xiii

Chapter 1: Drivers for Change 1

Driver: A New Product 1

Driver: Oracle Costs and Business Practices 5

Chapter 2: Requirements and Risk Assessment 15

Requirements of a New DBMS 16

Audits and Compliance 18

Risks 29

Chapter 3: Database and Application Code 33

Migrating the Database 34

Database Side Programming 37

Application Code 43

Becoming Database Agnostic 45

Chapter 4: Making the Decision 49

Extolling the Benefits 50

Presenting Cost Savings 51

Develop a Strategy 54

Putting it Together 56

Chapter 5: MariaDB Solution 59

Preinstallation Considerations 60

MariaDB Stand-Alone with Replication 62

Replication Limits and Testing 69

Galera Clustering 69

Chapter 6: Change as a Catalyst 73

Evaluating Solutions for Rework 75

Fixing the Legacy 77

Standards Adoption 78

Process Improvement 81

Automation 84

Chapter 7: Defining a Roadmap for Success 89

Database Evaluation 90

Evaluating First Steps 92

Path of Least Resistance 94

Success 95

Chapter 8: Making the Data Move 97

First Steps 98

Letting DBAs be DBAs 99

Tool Building 101

FOPEN 102

FCLOSE 104

FFLUSH 104

PUT 104

PUT_LINE 105

PUT_RAW 106

Dynamic SQL 107

Handling LOB Data 115
Sample Solution Code 119

Appendix A: Open Source Continuum 135

Open Source in the Data Center 136
Entrepreneurial Limits of Big Name Proprietary Systems 138
Where Is Open Source Not Viable 142
Benefits of Open Source 143

Index 145

About the Author

William Wood is an IT professional who has worked across many disciplines in his 18-year career. He started his work experience as a student worker for his school's engineering department, which ran the school's website, doing LAMP Stack work for database-driven dynamic website development. He has been working specifically in the database field for the past 10 years, first on a team that supported development infrastructure and release engineering where he became the Oracle SME, and then entering the working database administrator field in 2010 as a DBA, primarily working with and supporting Oracle and Oracle RAC in high-volume, compute intensive, and high availability environments.

About the Technical Reviewer

Ben Stillman is Director of Subscriber Services at MariaDB Corporation. He has over ten years experience with MariaDB and MySQL, and prior to that had several years of experience with Oracle Database and SQL Server. Ben is well-versed in migration challenges and has a good understanding of the various database platforms on the market and what each brings to the table. Ben resides near Columbus, Ohio where he enjoys spending time with his family and riding motorcycles.

Introduction

Migrating to MariaDB covers a wide range of topics that can be applied to many facets of the information technology industry in that the same methods and practices can be used in any type of migration and development project. There are many approaches to tackling a monumental task, and presented here you will find the strategies and methodologies that I have adopted over many years in the technology sector. This work follows a fictional company, FWP, and the leader of their database team, Vernon, through a database migration from Oracle to the MariaDB Open Source database.

The fictional portions of the story are based on many years spent in the technology sector as well as the educational endeavors that preceded them. The name of the company, FWP, was chosen entirely in jest for a comical take on what many will recall from their educational experience with the work examples and problems that all seemed to relate to one kind of widget or another. I grew to detest the widget and all that it stood for, so using it so widely says a bit about my acerbic sense of humor and wit. The story itself spans experiences and observations as seen throughout my career, to add a bit of storyline to what is many times considered the dry topic of technical information. It also provide a vehicle for explaining the how and why of many things that have been accomplished in a varied career, highlighting the successful migration from the Oracle database to the MariaDB database.

The more technically oriented reader may gravitate to a few chapters, while the project planners and managers might glean more by reading it in its entirety. Whatever the reader's strategy, there are many gems that can be gleaned from each chapter no matter their discipline or background. The first chapters provide the storyline and background for a small

fictitious company that has a new solution that needs a more cost-effective solution for its database backend, and how the head of their database department went about making this transition successfully. Vernon and FWP are fictional, as are their solutions; however, I have made the same database migration successfully using the same methodologies, roadmap, and solutions as discussed here. This backend migration was completed successfully on many levels and is still ongoing at the time of this writing, which is part of why getting this completed was a struggle with the timeline and final product.

Many of the methodologies on display throughout this book have been around for a very long time. One of my favorites that seems to display itself time and time again is that of the age-old mantra of "Keep It Simple, Stupid", referred to as KISS. It is very much applicable today as it was twenty years ago as I was exploring my educational pursuits. This is something that I have seen overlooked so many times with overworked, over-obfuscated, and increasingly complex solutions for problems that could be solved in a much simpler manner, making the solution easier to maintain, support, and deploy. There are also many methodologies covered here that have been around for a long time and are enjoying a rebranding or a reemergence in more recent times. Hopefully KISS comes full circle as well, as I am a big fan of the simple and effective solution.

There are many ways that one could make this same migration, so the more important aspect of this work is the path to follow, and not get hung up in the solution as applied here. What worked well here may not be as efficient for a much larger entity with a much larger data set size; however, the roadmap will still be the same and the solutions as provided will work when modified to suit the tasks requirements. Using Oracle's own tool set in the migration of the data carried some weight, as these tools were already available within the database software. This meant no additional cost for software to do this work, making the migration even more fiscally responsible for any entity undertaking a similar task, arbitrary of the database they may be migrating to with few modifications.

CHAPTER 1

Drivers for Change

There are many drivers for change in the world of technology and business. We are going to look at a couple of those in the following chapters from the viewpoint of a fictional company that has come out with a new product while at the same time going through a licensing audit. These two catalysts caused the company, Financial Widgets Plus (FWP), to evaluate their current database solution and possible alternative solutions because the cost, along with the overhead of use, of the proprietary solution could no longer be supported or fiscally responsible. They needed a replacement that would propel their new platform into the forefront while allowing them to generate more revenue to drive and support growth.

We will follow this fictional company, along with the fictional head of their database department, Vernon, as they go through the evaluation process to implementation. The hurdles as seen by FWP are identical to what any small software company would see when going through the evaluation and development of a migration plan for moving away from a high-cost, proprietary, closed database system like Oracle to an open solution, which in this case is MariaDB.

Driver: A New Product

The world of data and databases today is full of complex solutions and ever evolving buzzwords. However, nothing can be more confusing and daunting than the underlying costs, considerations, and licensing abstracts

W. Wood, *Migrating to MariaDB*, https://doi.org/10.1007/978-1-4842-3997-1_1

when considering a Database Management System (DBMS) or migrating one's topology from one solution to another. This can easily become compounded into a daunting undertaking, depending upon one's type of business and the requirements that lie within. In the following chapters and throughout this book we will be looking at a lot of the decisions, requirements, and special considerations from the standpoint of a fictitious company (FWP) that falls within the financial sector of the business world. There are two main drivers behind FWP looking at an alternative database solution, a new product coming into fruition and trying to leverage its deployment in a cost-effective manner, so we will be diving into a majority of the aspects of these changes.

The new product that we will be talking about was the brainchild of a newly hired database administrator, Vernon, with FWP. This software company has been around for many years, offering a highly customizable product platform to large entities within the financial world. The company had done well for these many years offering this highly customized service to these large-scale lenders within the ever-evolving financial world. However, these larger entities were beginning to evolve as well and were starting to bring this exact type of service in-house, so the days were numbered for providing this highly customized financial widget platform that we will refer to as the Custom Financial Widget, or CFW moving forward. The CFW platform and methodologies were severely outdated. It was also starting to cut into profits because each one of the solutions was not sharing a common code base; no processes were done in the same manner twice; and it required dedicated resources for each implementation, requiring that someone had the background knowledge to keep it moving along. It did not take a rocket scientist to see that if FWP continued to operate in the same fashion, that its longevity was limited and at stake. Vernon was not a rocket scientist and he saw this, but he also saw the possibilities of effecting change to FWP, which he had come to work at with the attitude of it being the last job of his career prior to retirement.

Vernon knew that the task of moving away from the CFW product line was multilayered and would be no easy task. The company had a "customize" mindset that had to change, and was so ingrained that it was effectually an uphill battle to even get some ear time for this idea to get traction. When Vernon presented his idea to the first person, the response was "FWP doesn't want to grow, it does not want more customers, because there is too much money to be made doing what we do." This was definitely not a warm reception to be sure. It also had some undertones of management practices he had seen previously in his career many times before, so Vernon sat on this for the time being and contemplated, waiting for the right opportunity. As he waited for this opportunity, there started to be some rumblings about this new idea being shopped around at FWP for a turnkey Widget, something that could be quickly deployed, easy to support, etc. It's amazing how that works in the business world, and this could very well be a topic for another book; however, we will go back to Vernon's next step trying to get this idea to fruition.

A few months later, while having a family dinner out with the General Manager (GM) of FWP, Vernon seized the opportunity to explain the full ramifications and scope of his idea. This involved developing a standardized engine for a new product line called Standardized Financial Widgets, which we will refer to here as SFW, with an easily repeatable and common code base as the heart of the product. Then all the best parts of the current CFW could be rolled into plug and play modules if you will, having a multi-tiered approach. For example, if a customer wanted to be able to have electronic data warehousing reports, then that was a pluggable module, and with some of the more advanced modules the customer could move up to the next tiers. The other methodology for tiering would be through the number of transactions; if a customer did not plan to process enough Widgets to make it fiscally conceivable, then they would have to pay more for the base service. Or if they wanted system of record long-term storage of their Widgets, that could be done as well, but

also with an upcharge for the storage requirements. Suddenly we were talking about a viable solution that could target both large customers and small, along with everything in between. The beauty of this solution, and what Vernon thought was the biggest selling point, was that not only would it generate revenue, but also lower the risk to FWP. That's because, being able to target smaller customers, they would be reducing the impact of losing a customer due to circumstances beyond their control, and this really obtained the effect that Vernon was looking for with the GM of FWP.

The new SFW product really started to get some momentum after this, and the GM requested that Vernon be the database administrator (DBA) assigned to this new venture. Interestingly, this did not pan out for Vernon as he had hoped. Even though the first couple of meetings seemed to go well and he contributed some really good ideas on how to proceed from the database side of things, he was then removed from the project after the third week by none other than the same boss who said "FWP does not want to grow." There are battles throughout life and careers and so Vernon decided to bide his time even though this was a major setback for him. It was okay because the idea for SFW continued to move forward, although slowly and not without its hurdles, and a somewhat abstract portion of the concept came into being with the first few customers. This proved the logic and marketability of the new product; however, limitations started to be seen, with the biggest one being the current DBMS solution that FWP had been using for over ten years. It was a solid foundation once Vernon went to work starting to improve and slim down the footprint into a more stable, fast, and lean deployment. In addition, Vernon began taking a very proactive approach to database principles that previous administrators had overlooked or just never considered. There was only one problem, scalability, both fiscal and physical resource, as the DBMS of choice from a historical perspective for FWP was Oracle Enterprise Edition with Real Application Clusters (RAC) and Advanced Security Option (ASO).

Driver: Oracle Costs and Business Practices

Like many organizations in the financial sector, such as banking institutions, credit card providers, and mortgage companies, FWP built their digital footprint around the tried and true architectural solutions of the time. For the most part these solutions oriented around a DBMS running on System V UNIX variants like Solaris, HPUX, and IBM's AIX. However, luckily FWP had already initiated conversions away from the old System V Unix variants and IA64 architecture, choosing to adopt RedHat Enterprise Linux in its stead along with moving away from the old Itanium-based servers to newer and faster machines based on the x64 architecture. The Oracle DBMS had become the solution of choice, especially with the combination of Real Application Clusters for hardware failover and the ASO (Advanced Security Option) for encryption of data at rest providing a secure and robust solution for any organization that deals with data protection requirements. During the proof of concept for these deployments, Vernon ran some pretty intensive stress tests against Oracle RAC on the newer architecture with the expected results of the combination performing superior to the previous and outdated architecture. However, this notwithstanding, the Oracle solution came at a very steep price that has grown significantly over the years, as has the complexity of licensing these solutions due to the advent of constantly evolving technologies such as virtualization, hardware partitioning, etc. that have and continue to evolve at an accelerated rate.

One cannot fault Larry Ellison for coming up with the licensing errata as instituted by the Oracle Corporation, as this was an absolutely brilliant idea. All one had to do was look at the basics of Moore's Law to see that as the architecture and solutions grew at seemingly exponential rates, thus would the coffers of Oracle. One of the aspects about the Oracle DBMS that helped solidify it as a revenue generating machine was the proprietary solutions that it offered to solve many complex problems with built in

capabilities, optimization engines, and fault tolerance that other vendors did not have. The only standard Oracle was willing to adhere to was their own; while many other vendors worked on and solidified standardizations like SQL-99, Oracle did everything their way. The result is a fantastically stable high-performance DBMS solution that works so well that many of its customers and end users shudder to think of what the results would be in migrating to anything else. So they continue to pay the exorbitant yearly price tag associated with what used to be the only high-grade solution on the market with the requirements for high-volume transactional processing in a high availability always environment. The mere thought of having to migrate large volumes of database structure and data, especially considering all the built-in functionality that may have been used with application code side solutions that were driven by the back-end database, is daunting. It was a monumental task that Vernon and his team chose to undertake, due to but not limited to, the following major points:

- The high cost of the Oracle solution(s)
- Having to run a mission critical DBMS on outdated hardware, because if they upgraded to a more powerful architecture with more internal processors, the costs incurred would be significant
- The Oracle pricing model does not fit a small to mid-sized company.
    - Buy Oracle products in quantity, then the pricing is much cheaper
- Customers cannibalizing revenue-generating audit approaches being employed over the last few years
- The sales approaches that open one up to be cannibalized