Synthesis Lectures on Data Management

SYNTHESIS
COLLECTION OF TECHNOLOGY

Alberto Lerner · Philippe Bonnet

Principles of Database and Solid-State Drive Co-Design

Springer

Synthesis Lectures on Data Management

Series Editor

H. V. Jagadish, University of Michigan, Ann Arbor, MI, USA

This series publishes lectures on data management. Topics include query languages, database system architectures, transaction management, data warehousing, XML and databases, data stream systems, wide scale data distribution, multimedia data management, data mining, and related subjects.

Alberto Lerner · Philippe Bonnet

Principles of Database and Solid-State Drive Co-Design

Springer

Alberto Lerner
Computer Science Department
University of Fribourg
Fribourg, Switzerland

Philippe Bonnet
Department of Computer Science
University of Copenhagen
Copenhagen, Denmark

ISSN 2153-5418 ISSN 2153-5426 (electronic)
Synthesis Lectures on Data Management
ISBN 978-3-031-57876-2 ISBN 978-3-031-57877-9 (eBook)
https://doi.org/10.1007/978-3-031-57877-9

Foreword

When I acquired my first hard disk drive in 1988, it was a modest 5.25-inch disk with a 30 MB capacity. Looking back, neither its storage capacity nor its performance was remarkable, but compared to the floppy diskettes of that era, it felt like a leap into a new realm. Over time, the performance of computer systems has advanced at an astonishing pace. Hard disk drives (HDDs), which have been a cornerstone of data storage for decades, are swiftly making way for solid-state drives (SSDs). SSDs stand out as a fascinating storage solution for several reasons. They offer exceptional performance and substantial storage capacities. The high-performance feature of SSDs can be attributed to the advent of the NVMe interface, purpose-built for SSDs, and the parallel operation of multiple flash memory components. NVMe not only facilitates rapid data transfers but also unlocks new horizons for storage capabilities, significantly enhancing overall computer system performance and introducing features that were once elusive with HDDs, such as computational storage.

At present, I am involved in SSD development at a memory semiconductor company, working on diverse SSD types, including those designed for servers, PCs, and laptops. With each generation of interfaces, SSDs continue to push the boundaries of capacity and performance. Moreover, SSDs can be tailored to harmonize seamlessly with server and computer systems, unlocking numerous additional features and performance enhancements. Nevertheless, despite most storage products efficiently fulfilling their roles, there is still limited progress in optimizing performance and functionality through vertical integration.

Vertical optimization necessitates a profound comprehension of both the host system and the storage device. While the roles of the host system and the storage device remain distinct, the richness of the semantics supported by interfaces has unlocked the potential for vertical optimization. Surprisingly, there is only a limited body of literature exploring software and hardware layers from applications to devices, especially concerning interactions related to database applications prevalent in servers and data centers.

This is where Philippe and Alberto, two researchers whom I deeply admire, come into the picture. They are diligently crafting a book that addresses precisely these topics. Their substantial body of work in storage vertical optimization has left an indelible mark, inspiring me to delve into their papers and engage in research with my students. It is with immense pleasure and a sense of honor that I write this message after finally getting a glimpse of their draft.

Philippe and Alberto are wonderful advocates for open-source SSD, called OpenSSD, and have been exceptional mentors. When I embarked on my journey with OpenSSD alongside my university students, my initial motivation was to contribute to their remarkable efforts. Despite the myriad challenges they faced, they have made remarkable contributions to OpenSSD, presenting their findings at prestigious conferences and journals.

As we stand on the cusp of a new era in storage interfaces, one that promises to usher in a wave of groundbreaking storage devices, the landscape of data access is on the brink of transformation. These new storage devices are poised to deliver lightning-fast data access performance for data centers and AI applications, paving the way for novel vertical optimizations. Traditional block devices are evolving to embrace byte device characteristics, offering enhanced data access efficiency. This efficiency surge is empowered by the storage device's deeper understanding of the host system.

I eagerly anticipate this book's publication for the fresh insights it provides and how collaborations with my colleagues to expedite product development will benefit. I trust you will find this book to be a source of enlightenment about the present and future of storage.

Seoul, South Korea
October 2023

Yong Ho Song

Preface

The motivation to write this book started perhaps some ten years ago. At that time, Philippe had his Research Statement posted online, and, thanks to one of those coincidences that life is full of, Alberto got a glimpse at it. Philippe had argued that storage devices and their stack should be more flexible than they were at the time. There were some ideas on how to use software—programmability, really—to put that flexibility into the hands of application writers. Right there, Philippe got Alberto's attention, even though neither of us realized it at the time. We have started a conversation that has lasted ever since and culminates in this book.

The book attempts to describe a Solid-State Drive as a layered system. We are far from the first to talk about SSDs' inner workings, but we felt that such information was too scattered. Certain layers are interfaces to devices and are commonly discussed in the context of File or Operating Systems. Other layers are firmware that live inside the device and are published at very specific storage venues. Lastly, some layers consist entirely of hardware and present idiosyncrasies that are less accessible to software programmers. By presenting the layers in a single place, we hope to give the reader a better chance at comprehending a typical device's behavior.

However, consolidating information was but an initial goal. We also wanted to formulate and substantiate what can be seen as the book's central claim: SSDs are malleable. There should be no such thing as a standard behavior for this class of device. Our premise is that if we understand what an application wants from a device, we can (increasingly easily) tailor it to meet these needs. By doing so, we can achieve much better performance and efficiency throughout the entire system. We say that **SSDs and the applications using them can be co-designed**. This is no small claim; generations of SSD users have been trained to unilaterally adapt their applications to how a mythical standard SSD works, never to try to decide how the device and the application may, together, accomplish a task.

But to which applications should we tailor SSDs? In the book, we focus on data-intensive applications, such as database systems, for two reasons. First, we are familiar with these systems, having helped build several of them throughout our careers. Second, these systems generate a rich set of workloads that can be easily characterized. If we know a workload, that is, the patterns and sensitivities that an application exposes while performing I/Os, we can alter a traditional SSD structure to better adapt to these.

Therefore, the core of the book lies in the chapters that describe such alterations. Indeed, we created a classification of techniques to change devices. Chapter 4 describes how to do so by attaching a computational element between an SSD and an application. Neither the application nor the device is changed internally, but now the application can issue requests beyond fetching and storing data that the computational element will carry out.

Chapter 5 discusses how to affect behavioral changes in an entirely different way. It swaps specific subsystems within an SSD so as to perform tasks in a way that benefits the target workloads. In other words, in contrast to the external changes made in computational devices, the changes here are all internal, and no new element is introduced to the device.

Chapter 6 takes a hybrid approach to customizing devices. It mixes the techniques seen in the previous two chapters by allowing both internal component changes and additions. These three chapters introduce a new taxonomy through which we study the different techniques. We tried to fill the discussions in these chapters with actual device examples, both commercial and academic.

There are many ways for readers to prepare before getting to those core chapters. Those already exposed to SSD internals, the OS storage stack details, and the interface between both could warm up in the introduction presented in Chapter 1 and jump straight into one of the core chapters. If we mention concepts in those chapters introduced elsewhere, we try to leave links connecting the two.

For the beginner or intermediate readers, we recommend looking into Chapters 2 and 3 after the Introduction. They discuss layers that will come in handy when talking about the different ways we believe devices could be co-designed along with the applications that use them. For the curious reader, regardless of their level, we tried to get to the very fundamentals of Flash memory in Appendix A. We also paint a portrait of an SSD as an orchestration of several algorithms in Appendix B.

Lastly, we humbly feel that this book is in its infancy and hope that, if it helps other researchers and practitioners, we could improve it in a future edition. Your feedback as a reader would go a long way. In the meantime, we sincerely hope you enjoy the book. We certainly had a blast writing it.

Fribourg, Switzerland — Alberto Lerner
Copenhagen, Denmark — Philippe Bonnet
January 2024

Acknowledgments

We are indebted to Yong Ho Song and to the many students and collaborators who have helped shape our understanding of the field. We would like to name in no particular order: Sangjin Lee, Kibin Park, Jinwoo Jeong, Luc Bouganim, Björn Þór Jónsson, Niclas Hedam, Matias Björling, Simon Lund, Ivan Luiz Piccoli, and Javier Gonzalez.

Alberto received funding from the Swiss State Secretariat for Education (SERI) in the context of the SmartEdge EU project (grant agreement No. 101092908). Philippe received funding from the European Union’s Horizon 2020 Research and Innovation Programme (grant agreement No. 957407/Daphne).

Contents

About the Authors

Alberto Lerner has a mixed industrial and academic profile with over 30 years of experience. He was a research or software technical staff at numerous tech companies, such as IBM, Google, and MongoDB, or a consultant at many database start-ups. His interest revolves around high-scale, high-performance distributed systems, particularly using heterogeneous hardware to support them. He has participated in designing and implementing several such systems, including, more recently, the `X-SSD` device and ongoing efforts to create more easily programmable co-designed devices.

Alberto has been a Senior Researcher at the Computer Science Department of the University of Fribourg in Switzerland since 2018. He has been on several Program Committees for the Database and Systems communities, including SIGMOD, VLDB, CIDR, EDBT, ICDE, and Usenix ATC.

Philippe Bonnet is an experimental computer scientist with a background in database systems. For 30 years, Philippe has explored the design, implementation, and evaluation of database systems in the context of successive generations of computer classes, including wireless sensor networks, computer clusters, and most recently computational storage. Philippe is an expert on storage system software. He contributed to the uFlip Benchmark, the Linux multi-queue block layer, the Linux framework for Open-Channel SSDs, the OX architecture for computational storage, the xNVMe library, and Delilah, a prototype for eBPF offload on computational storage.

Philippe is a professor at DIKU, the Department of Computer Science of the University of Copenhagen. He was faculty at the IT University from 2009 to 2023. He is a trustee of the VLDB Endowment and currently chairs the ACM EIG on Reproducibility and Replicability.