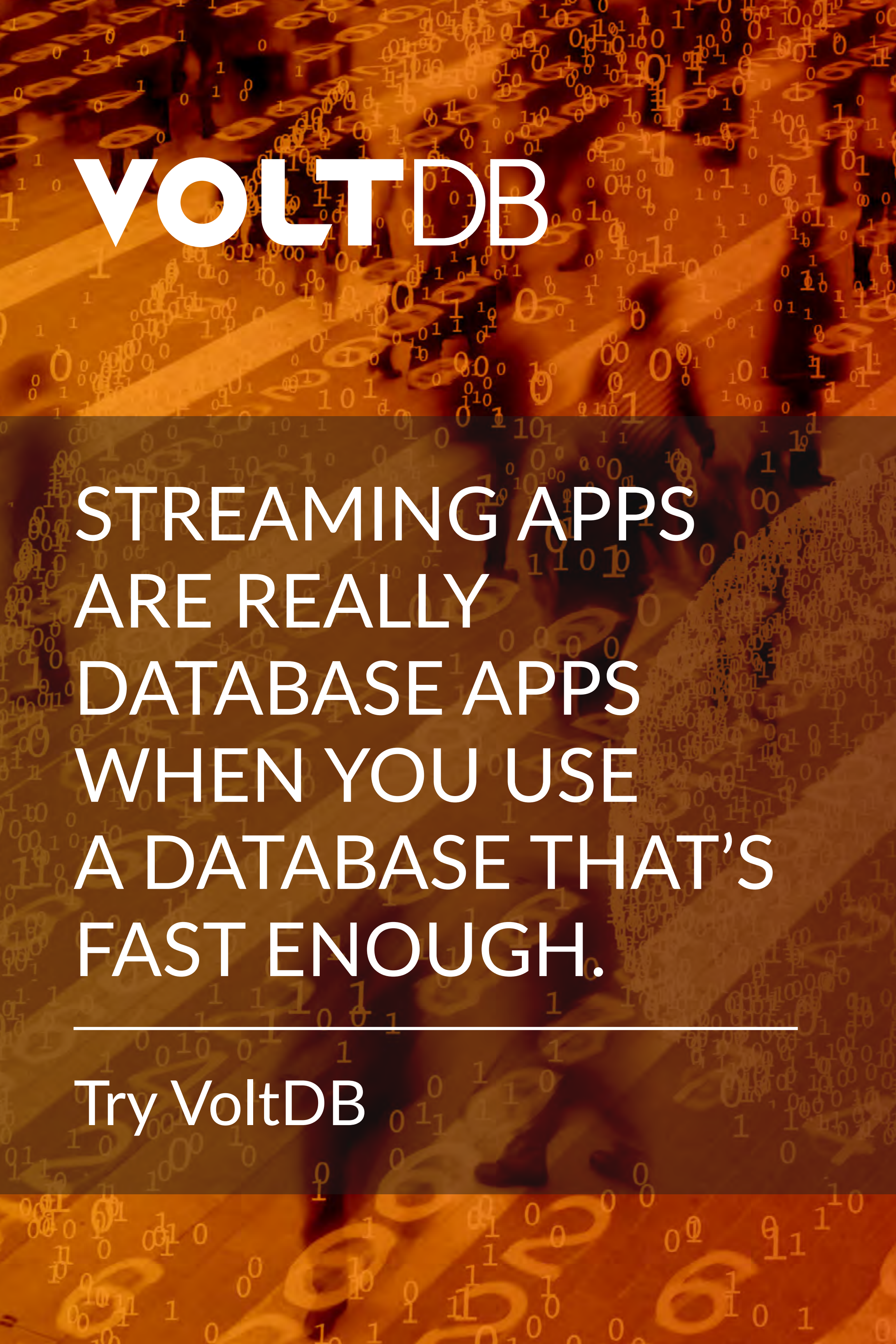
O'REILLY®

Compliments of VOLTDB

Fast Data and The New Enterprise Data Architecture

Scott Jarr


VOLTDB
STREAMING APPS ARE REALLY DATABASE APPS WHEN YOU USE A DATABASE THAT'S FAST ENOUGH.
Try VoltDB

Fast Data and the New Enterprise Data Architecture

Scott Jarr

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'REILLY®

Fast Data and the New Enterprise Data Architecture

by Scott Jarr

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Editor: Jenn Webb **Illustrator:** Rebecca Demarest

October 2014: First Edition

Revision History for the First Edition:

2014-09-24: First release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491913888* for release details.

While the publisher and the author(s) have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author(s) disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

ISBN: 978-1-491-91388-8

[LSI]

Table of Contents

Preface

A structural shift in data management is underway. Unlike previous eras of technological change—mainframe to server, server to PC, PC to mobile and tablet—this shift is not driven solely by growth in processing power (the oft-cited Moore's Law). Today, processing power is cheap at the endpoints. The combination of cheap, ubiquitous CPUs attached to fast mobile networks is creating a network effect of devices, distorting Moore's Law with the force multiplier of near-global wireless network coverage. Thus, today's shift is spurred not only by increases in processing power but also by the growth of data—of *new* data, which is doubling every two years—and by the rate of growth in the *perceived value* of data.

These macro computing trends are causing a swift adoption of new data management technologies. Open source software solutions and innovations such as in-memory databases are enabling organizations to reap the value of realtime interactions and observations. No longer is it necessary to wait for insight until the data has been analyzed deeply in a big data store. This is changing the way in which enterprises manage data, both data in motion--"fast data" streaming in from millions of endpoints—and data at rest, or "big data" stored in Hadoop and data warehouses.

Businesses in the vanguard of this change recognize that they operate in a "data economy." These leaders make an important distinction between the two major ways in which they interact with data. This shift in thinking has led to the creation of a new enterprise data architecture. This book will discuss what the new enterprise data architecture looks like as well as the benefits it will deliver to organizations. It will also outline the major technology components necessary to build a unified enterprise data architecture, one in which both fast data and big data work together.

CHAPTER 1

What's Shaping the Environment

Data Is Everywhere

The digitization of the world has fueled unprecedented growth in data, much of it driven by the global explosion of mobile data sources and the Internet of Things (IoT). Each day, more devices—from smartphones to cars to electric grids—are being connected and interconnected. It is safe to predict that within the next 10–15 years, anything powered by electricity will be connected to the Internet.

According to the 2014 EMC/IDC Digital Universe report, data is doubling in size every two years. In 2013, more than 4.4 zetabyes of data had been created; by 2020, the report predicts that number will explode by a factor of 10 to 44 zetabytes—44 trillion gigabytes. The report also notes that people—consumers and workers—created some two-thirds of 2013's data; in the next decade, more data will be created by things —sensors and embedded devices. In the report, IDC estimates that the IoT had nearly 200 billion connected devices in 2013 and predicts that number will grow 50% by 2020 as more devices are connected to the Internet—smartphones, cars, sensor networks, sports tracking monitors, and more.

Data from these connected devices is fueling a data economy, creating huge implications for future business opportunity. Additionally, the rate of growth of new data is creating a structural change in the ways enterprises, which are responsible for more than 80% of the world's data, manage and interact with that data.

As the data economy evolves, an important distinction between the major ways in which businesses interact with data is emerging. Companies have begun to interact with data that is *big*—data that has volume and variety. Additionally, as companies embark on ever-more extensive big data initiatives, they have also realized the importance of interacting with data that is *fast*. The ability to process data immediately—a requirement driven by IoT macro-trends—creates new opportunitu to realize value via disruptive busuiness models.

To illustrate this point, consider the devices generating all this data. Some are relatively dumb sensors that generate a one-way flow of information—for example, network sensors that push data to a processing hub but that cannot communicate with one another. More important are two-way sensors embedded in "smart" devices—for example, automotive in-vehicle infotainment and navigation systems and smart meters used in smart power grids. These two-way sensors not only collect data but also enable organizations to analyze and make decisions on that data in real time, pushing results (more data) back to the device. These smart sensors create huge streams of fast, smart data; they can act autonomously on "your" inputs as well as act collectively on the group's inputs.

The EMC/IDC report states that "embedded systems—the sensors and systems that monitor the physical universe—already account for 2% of the digital universe. By 2020 that will rise to 10%." Clearly, two-way sensors that generate fast and big data require different modes of interaction if the data is to have any business value. These different modes of interaction require the new capabilities of the enterprise data architecture.

Data Is Fast Before It's Big

It is important to note that the discussion in this book is contained to what are described as "data-driven applications." These applications are pervasive in many organizations and are characterized by utilization of data at scales previously unobtainable. This scale can refer to the complexity of the analysis, the sheer amount of data being managed, or the velocity at which data must be acted upon.

Simply stated, data is *fast* before it is *big*. With the increase in fast data comes the opportunity to act on fast and big data in a way that creates the most compelling vision for data-driven applications.

Fast data is a new opportunity made possible by emerging technologies and, in many cases, by new approaches to established technologies, e.g., in-memory databases. In the new paradigm—one in which data in motion has equal or greater value than “historical” data (data at rest) —new opportunities to extract value require that enterprises adopt new approaches to data management. Many traditional database architectures and systems are incapable of dealing with fast data’s challenges.

As a result, the data management industry has been enveloped in confusion, much of it driven by hype surrounding the major forces of big data, cloud, and mobility. Fortunately, many of the available technologies are falling into categories based on problems they address, bringing the picture into better focus. This is good news for application developers, as advances in cloud computing and in-memory database architectures mean familiar tools can be used to tackle fast data.

CHAPTER 2

The Enterprise Data Architecture

Introduction

The enterprise data architecture is a break from the traditional siloed data application, where data is disconnected from the analytics and other applications and data. The enterprise data architecture supports fast data created in a multitude of new end points, operationalizes the use of that data in applications, and moves data to a "data lake" where services are available for the deep, long-term storage and analytics needs of the enterprise. The enterprise data architecture can be represented as a data pipeline that unifies applications, analytics, and application interaction across multiple functions, products, and disciplines (see Figure 2-1).

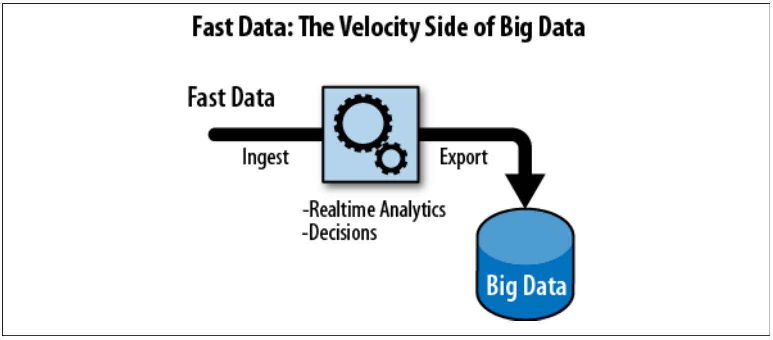
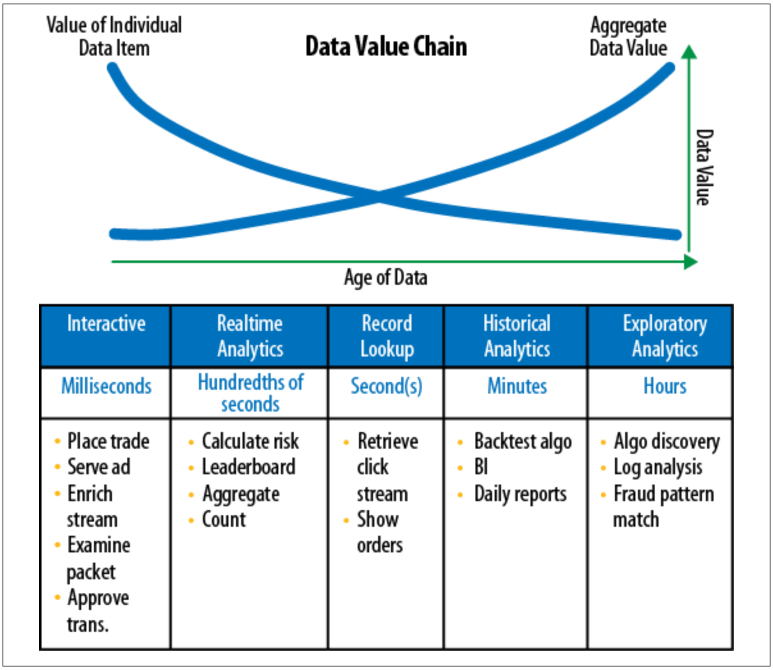


Figure 2-1. Fast data represents the velocity aspect of big data.

Data and the Database Universe

Key to understanding the need for an enterprise data architecture is an examination of the “database universe” concept, which illustrates the tight link between the age of data and its value.

Most technologists understand that data exists on a time continuum; it is not stationary. In almost every business, data moves from function to function to inform business decisions at all levels of the organization. While data silos still exist, many organizations are moving away from the practice of dumping data in a database—e.g., Oracle, DB2, MSSQL, etc.—and holding it statically for long periods of time before taking action.



Interactive	Realtime Analytics	Record Lookup	Historical Analytics	Exploratory Analytics
Milliseconds	Hundredths of seconds	Second(s)	Minutes	Hours
• Place trade • Serve ad • Enrich stream • Examine packet • Approve trans.	• Calculate risk • Leaderboard • Aggregate • Count	• Retrieve click stream • Show orders	• Backtest algo • BI • Daily reports	• Algo discovery • Log analysis • Fraud pattern match

Figure 2-2. Data has the greatest value as it enters the pipeline, where realtime interactions can power business decisions, e.g., customer interaction, security and fraud prevention, and optimization of resource utilization.

The actions companies take with data are increasingly correlated to the data’s age. Figure 2-2 represents time as the horizontal axis. To the far left is the point at which data is created. Immediately after data is created, it is highly interactive *and for each event, of greatest value*. This is where the opportunity exists to perform high-velocity operations on "new" or "incoming" data—for example, to place a trade, make a recommendation, serve an ad, or inspect a record. This is the beginning of a data management pipeline.

Shortly after data enters the pipeline, it can be examined relative to other data that has also arrived recently, e.g., by examining network traffic trends, composite risk by trading desk, or the state of an online game leader board.

> *Queries on fresh data in motion are commonly referred to as "realtime analytics."*

As data begins to age, the nature of its value begins to change; it becomes useful in a historical context and relative to other sources of data. For example, knowing a buyer's preference *after* a purchase is clearly less valuable than it would be *during* the purchase.

Organizations have found countless ways to gain valuable insights, such as trends, patterns, and anomalies, from data over long timelines and from multiple sources. Business intelligence and reporting are examples of what can be done to extract value from historical data. Additionally, big data applications are increasingly used to explore historical data for deeper insights—not just observing trends, but discovering them. This can be thought of as "exploratory analytics."

With the adoption of fast and big data technologies, a trend is emerging in the way data management applications are being architected, designed, and developed. A central tenet underlies modern data architecture design:

> *The value in data is not purely from historical insights.*

There is a natural push for analytics to be visible closer and closer to real time. As this occurs, it becomes obvious that taking action on this information, in real time, the instant it is created, is the ultimate goal of an enterprise data architecture. As a result, the historically separate functions of the "application" and the "analytics" begin to merge.

Enterprises are examining how they build new applications and new analytics capabilities. This natural progression quickly takes people to the point at which they realize they need a unifying architecture to serve as the basis for how data-heavy applications will be built across the company, encompassing application interaction all the way through to exploratory analytics. What has changed is that application interactions are now part of the pipeline. The result of this work is the modern enterprise data architecture.

Architecture Matters

Interacting with fast data is a fundamentally different process than interacting with big data that is at rest, requiring systems that are architected differently. With the correct assembly of components that reflect the reality that application and analytics are merging, an enterprise data architecture can be built that achieves the needs of both data in motion (fast) and data at rest (big).

Building high-performance applications that can take advantage of fast data is a new challenge. Combining these capabilities with big data analytics into an enterprise data architecture is increasingly becoming table stakes. But not everyone is prepared to play.

CHAPTER 3

Components of the Enterprise Data Architecture

Figure 3-1 illustrates the main components of an enterprise data architecture. The architectural requirements of the separation of fast and big are evident, with the capabilities and requirements of each presented.

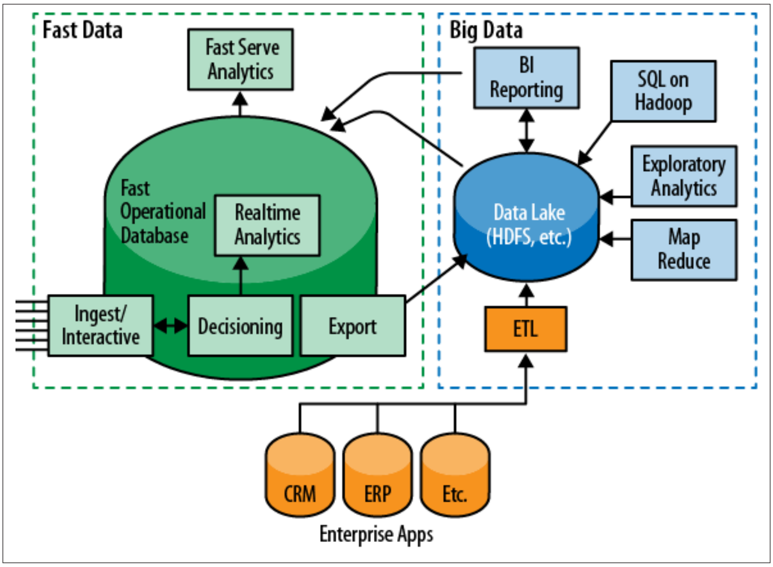


Figure 3-1. Note the tight coupling of fast and big, which must be separate systems at scale.

The first thing to notice is the tight coupling of *fast* and *big*, although they are separate systems; they have to be, at least at scale. The database system designed to work with millions of event decisions per second is wholly different from the system designed to hold petabytes of data and generate extensive historical reports.

Big Data, the Enterprise Data Architecture, and the Data Lake

The big data portion of the architecture is centered around a data lake, the storage location in which the enterprise dumps *all* of its data. This component is a critical attribute for a data pipeline that must capture all information. The data lake is not necessarily unique because of its design or functionality; rather, its importance comes from the fact that it can present an enormously cost-effective system to store everything. Essentially, it is a distributed file system on cheap commodity hardware.

Today, the Hadoop Distributed File System (HDFS) looks like a suitable alternative for this data lake, but it is by no means the only answer. There might be multiple winning technologies that provide solutions to the need.

The big data platform's core requirements are to store historical data that will be sent or shared with other data management products, and also to support frameworks for executing jobs directly against the data in the data lake.

Refer back to Figure 3-1 for the components necessary for a new enterprise data architecture. In a clockwise direction around the outside of the data lake are the complementary pieces of technology that enable businesses to gain insight and value from data stored in the data lake:

Business intelligence (BI) – reporting

Data warehouses do an excellent job of reporting and will continue to offer this capability. Some data will be exported to those systems and temporarily stored there, while other data will be accessed directly from the data lake in a hybrid fashion. These data warehouse systems were specifically designed to run complex report analytics, and do this well.

SQL on Hadoop

Much innovation is happening in this space. The goal of many of these products is to displace the data warehouse. Advances have been made with the likes of Hawq and Impala. Nevertheless, these systems have a long way to go to get near the speed and efficiency of data warehouses, especially those with columnar designs. SQL-on-Hadoop systems exist for a couple of important reasons:

a. SQL is still the best way to query data

b. Processing can occur without moving big chunks of data around

Exploratory analytics

This is the realm of the data scientist. These tools offer the ability to “find” things in data: patterns, obscure relationships, statistical rules, etc. Mahout and R are popular tools in this category.

Job scheduling

This is a loosely named group of job scheduling and management tasks that often occur in Hadoop. Many Hadoop use cases today involve pre-processing or cleaning data prior to the use of the analytics tools described above. These tools and interfaces allow that to happen.

The big data side of the enterprise data architecture has, to date, gained the lion’s share of attention. Few would debate the fact that Hadoop has sparked the imagination of what’s possible when data is fully utilized. However, the reality of how this data will be leveraged is still largely unknown.

Integrating Traditional Enterprise Applications into the Enterprise Data Architecture

The new enterprise data architecture can coexist with traditional applications until the time at which those applications require the capabilities of the enterprise data architecture. They will then be merged into the data pipeline.

The predominant way in which this integration occurs today, and will continue for the foreseeable future, is through an extract, transform, and load (ETL) process that extracts, transforms as required, and loads legacy data into the data lake where everything is stored. These applications will migrate to full-fledged fast + big data applications in time (this is discussed in detail in Chapter 7).

Fast Data in the Enterprise Data Architecture

The enterprise data architecture is split into two main capabilities, loosely coupled in bidirectional communications— *fast* data and *big* data. The fast data segment of the enterprise data architecture includes a fast in-memory database component. This segment of the enterprise data architecture has a number of critical requirements, which include the ability to ingest and interact with the data feed(s), make decisions on each event in the feed(s), and apply realtime analytics to provide visibility into fast streams of incoming data.

The following use case and characteristic observations will set a common understanding for defining the requirements and design of the enterprise data architecture. The first is the fast data capability.

An End-to-End Illustration of the Enterprise Data Architecture in Action

The IoT provides great examples of the benefits that can be achieved with fast data. For example, there are significant asset management challenges when managing physical assets in precious metal mines. Complex software systems are being developed to manage sensors on several hundred thousand "things" that are in the mine at any given time.

To realize this value, the first challenge is to ingest streams of data coming from the sheer quantity of sensors. Ingesting on average 10 readings a second from 100,000 devices, for instance, represents a large ingestion stream of one million events per second.

But ingesting this data is the smallest and simplest of the tasks required in managing these types of streams. As sensor readings are ingested into the system, a number of decisions must be made against each event, as the event arrives. Have all devices reported readings that they are expected to? Are any sensors reporting readings that are outside of defined parameters? Have any assets moved beyond the area where they should be?

Furthermore, data events don't exist in isolation from other data that may be static or coming from other sensors in the system. To continue the precious metal mine example above, monitoring the location of an expensive piece of equipment might raise a warning as it moves outside an "authorized zone." However, that piece of location data requires additional context from another data source. The movement might be acceptable, for instance, if that machinery is on a list of work orders showing this piece of equipment is on its way to the repair depot. This is the concept of "data fusion," the ability to make contextual decisions on streaming *and* static data.

Data is also valuable when it is counted, aggregated, trended, and so forth—i.e., realtime analytics. There are two ways in which data is analyzed in real time:

1. A human wants to see a realtime representation of the mine, via a dashboard—e.g., how many sensors are active, how many are outside of their zone, what is the utilization efficiency, etc.
2. Realtime analytics are used in the automated decision-making process. For example, if a reading from a sensor on a human shows low oxygen for an instant, it is possible the sensor had an anomalous reading. But if the system detects a rapid drop in ambient oxygen over the past five minutes for six workers in the same area, it's likely an emergency requiring immediate attention.

Physical asset management in a mine is a real-world use case to illustrate what is needed from all the systems that manage fast data. But it is representative. The same pattern exists for Distributed Denial of Service (DDoS) detection, log file management, authorization of financial transactions, optimizing ad placement, online gaming, and more.

Once data is no longer interactive and fast moving, it will move to the big data systems, whose responsibility it is to provide reliable, scalable storage and a framework for supporting tools to query this historical data in the future. To illustrate the specifics of what is to be expected from the big data side of the architecture, return to the mining example.

Assume the sensors in the mine are generating one million events per second, which, even at a small message size, quickly add up to large volumes of stored data. But, as experience has shown, that data cannot be deleted or filtered down if it is to deliver its inherent value. Therefore, historical sensor data must move to a very cost-effective and reliable storage platform that will make the data accessible for exploration, data science, and historial reporting.

Mine operators also need the ability to run reports that show historical trends associated with seasonality or geological conditions. Thus, data that has been captured and stored must be accessible to myriad data management tools—from data warehouses to statistical modeling—to extract the analytics value of the data.

This historical asset management use case is representative of thousands of use cases that involve data-heavy applications.

CHAPTER 4

Why Is There Fast Data?

Fast Data Bridges Operational Work and the Data Pipeline

While the big data portion of the enterprise data architecture is well designed for storing and analyzing massive amounts of historical data at rest, the architecture of the fast data portion is equally critical to the data pipeline.

There is good evidence, much of it evident in the EMC/IDC report's analysis of the growth in mobile, sensors, and IoT, that all serious data growth in the future will come from fast data. Fast data comes into data systems in streams; they are fire hoses. These streams look like observations, log records, interactions, sensor readings, clicks, game play, and so forth: things happening hundreds to millions of times a second.

Fast Data Frontier—The Inevitability of Fast Data

Clarity is growing that at the core of the big data side of the architecture is a fully distributed file system (HDFS or another FS) that will provide a central, commoditized repository for data at rest within the enterprise. This market is taking shape today, with relevant vendors taking their places within this architecture.

Fast data is going through a more fundamental and immediate shift. Understanding the opportunity and potential disruption to the status quo is beginning in earnest. Fast data is where many of the truly revolutionary advances will be made.

Make Faster Decisions; Don't Settle Only for Faster Analytics

In order to understand the change coming to the fast data side of the enterprise data architecture, one only needs to ask: "Why do organizations perform analytics in the first place?" The answer is simple. Businesses seek to make better decisions, such as:

- Better insight
- Better personalization
- Better fraud detection
- Better customer engagement
- Better freemium conversion
- Better game play interaction
- Better alerting and interaction

These interactions are the responsibility of the application, and the most valuable improvements come when these interactions are specific to the context of each event (i.e., use the current state of the system) and occur in real time.

Applications and Analytics Merge

Applications are the main point of entry for data streaming into the enterprise. They are the initial collection point and are responsible for the interactions discussed above. Application interaction has the same characteristics as described for fast data—ingest events, interact with data for decisions, and use realtime analytics to enhance the experience. The application is increasingly becoming both the organization's and the consumer's "interface" to the data.

However, this model is different from the historical way in which applications were developed. Before the dawn of the data-driven world, applications were written with an operational database to manage data interaction. Throughput requirements were low, and developers rarely worried about how analytics would be performed. Analytics was a secondary process. At some point after the application processed the data, it would be moved from the operational system into an analytics system, and someone other than the application developer would run and manage those analytics.

But application developers now realize applications must interact in real time with fast streams of data and use the analytics derived to make valuable interactions. The stresses this places on the fast data architecture necessitates new approaches that can meet these needs.

Progression to Realtime Analytics Necessitates Automated Decisions

Further pushing the adoption of fast data within the enterprise data architecture are the widespread—and divergent—expectations about what can be achieved with analytics.

Years ago, it could take a month to collect data and generate analytics. Data would be collected, a report would be run, and the report handed to a business user. The user would evaluate the analysis and make a decision on a course of action. The human-driven decision process often would take hours to make.

With the advent of modern data warehouses and faster technology, that reporting process has declined over the years to hours or even minutes. But the human portion of the decision process hasn't changed, still requiring some number of hours or minutes to make. Fast reporting provided a tangible benefit by making analyses available to people faster, but did little to speed the human decision-making process.

But businesses have a very strong desire to get to realtime visibility into their operations, especially those that involve fast-moving data. Quickly after the decision to enable realtime analytics occurs, the challenges of the human decision process come to light. The speed at which decisions now take place has quickly surpassed the speed a human can handle. These speeds necessitate automated decisions on actionable analytics.

CHAPTER 5

Requirements of Fast Data Systems in the Enterprise Data Architecture

Interaction is what the application is responsible for, and the most valuable improvements come when one can do these interactions accurately and in real time. This brings us to the fast data segment of the enterprise data architecture, where we deal with fast data to make better, faster realtime applications.

> *Systems designed to handle fast data must merge the capabilities of three historical products: operational databases, realtime analytics, and stream processing.*

Many solutions are emerging in the fast data market from players like Amazon and Google, a testament to the fact that a data problem is looming. Unfortunately, by focusing mainly on stream processing, these solutions miss a huge part of the value organizations can gain from fast data. Fast data is a new frontier. It is an inevitable step organizations will take when they begin to deeply integrate analytics into the data management architecture. Committing to a path that does not address all these capabilities ensures an organization will be rewriting its systems far sooner than desired.

As data has become more immediately valuable, application developers have realized applications now need to interact with fast streams of data *and* analytics to take advantage of the data available to them. This recognition surfaces the requirements of fast data.

Building an Architecture for Fast Data

What's needed to build a data-driven application that runs on streams of *fast* and *big* data? It comes down to three general requirements to get it right. Some requirements are negotiable, but any decision to waive a requirement should be driven by the application's needs, not by a limitation of the data management technology chosen.

The requirements of fast data applications are covered in the following sections.

1. Ingest/interact with the data feed

Much of the interesting data coming into organizations today is coming fast, from more sources, and at a greater frequency. These data sources are often the core of any data pipeline being built. However, ingesting this data alone isn't enough. Remember, there is an application facing the stream of data, and the "thing" at the other end is usually looking for some form of interaction.

For example, VoltDB, an in-memory NewSQL database, is powering a number of smart utility grid applications, including a rollout of 53 million meters. With these numbers of meters outputting multiple sensor readings per second comes a serious data ingestion challenge. Moreover, each reading needs to be examined to determine the status of the sensor and whether interaction is required.

2. Make decisions on each event in the feed

Using other pieces of data—previous events, events from other endpoints, static data—to make decisions on how to respond enhances the interaction described previously; it provides much-needed context to decisions. Some amount of stored data is required to make these decisions. If an event is taken only at its face value, the architecture is missing the context—the current state—in which the event occurred. The ability to make better decisions because of things you know about the *entire* application is lost.

Consider this example: Utility sensor readings become much more informative and valuable when one can compare a reading from a single meter to 10 others connected to the same transformer. This makes it possible to determine if there is a problem with the transformer, rather than inferring the problem lies with a single meter located at a home.

The following example might strike closer to home. A family is online shopping for flat-screen TVs. If they are presented with recommendations for what other shoppers purchased when they bought TVs, the recommendation would be timely, but not necessarily relevant—i.e., the recommendation system doesn't know if they are buying a TV for a child's room or a TV for a family room. Thus, if they are provided with recommendations based on aggregated purchase data, those recommendations will be relevant, but may not be personalized.

Recommendations need *context*—or state—to be relevant, they need to be *timely* to be useful, and they need to be *personalized* to the shopper's needs. To accomplish all three—and to do it without tradeoffs—businesses need to act on each event, with the benefit of context, i.e., stateful, stored data. The ability to interact with the ingest/data feed means businesses can know what the customer wants, at the exact moment of his or her need.

3. Provide visibility into fast-moving data with realtime analytics

The ability to make sense of fast-moving data extends beyond the capabilities of a human looking at a dashboard. One thing that makes fast data applications distinguishable from old-school online transaction processing (OLTP) is that realtime analytics are used *in* the decision-making process. By running these analytics within the fast data engine, operational decisions are informed by the analytics. The ability to take more than just a single event into context when making a decision makes that decision much more informed. In big data, as in life, context is everything.

Returning to the smart meter example, transformers show a particular trend prior to failure, and failure of that type of electrical componentry can be rather spectacular. It's important to identify these impending failures before they happen. This is a classic example of a realtime analytic that is injected into a decision-making process. IF a transformer's 30 minutes of prior data indicate it is TRENDing like THIS, THEN shut it down and reroute power.

In addition, fast data provides two more functions critical to the enterprise data architecture, described in the following two sections.

Fast data systems must seamlessly integrate into systems designed to store big data

One size does not fit all when it comes to database technology in the 21st century. So, while a fast in-memory operational database is the correct tool for the job of managing fast data, other tools are optimized for the storage and deep analytic processing of big data. Moving data between these systems is an absolute requirement.

However, much more than just data movement is required. In addition to the pure movement of data, the integration between big data and fast data must allow:

- Dealing with the impedance mismatch between the big system's import capabilities and the fast data arrival rate.
- Reliable transfer between systems, including persistence and buffering.
- Pre-processing of data, so when it hits the data lake it is ready to be used.

In the smart grid example, fast data coming from smart meters across an entire country accumulates quickly. This historical data has obvious value in showing seasonal trends and year-over-year grid efficiencies. Moving this data to the data lake is critical. But validations, security checks, and data cleansing can be done prior to the data arriving in the data lake. The more this integration is baked into data management products, the less code the application architect needs to figure out, e.g., how to persist data if one system fails and where to overflow data if the data lake can't keep up with ingestion rates.

Fast data systems must have the ability to serve analytic results and knowledge from big data systems quickly to users and applications, closing the data loop

The deep, insightful analytics generated by BI reports and analyzed by data scientists need to be operationalized, i.e., able to use realtime data. This can be achieved in two ways:

- Make the BI reports consumable by more people/devices than the analytics system can currently support
- Take the intelligence from the analytics and move it into the operational system

The first point is easy to describe. Reporting systems such as data warehouses and Hadoop-based systems do a great job generating and calculating reports. However, they are not designed to serve those reports to thousands of concurrent users with millisecond latencies. To meet this need, many enterprises are moving the results of these analytics stores to an in-memory operational database component that can serve these results at fast data's frequency/speed. We will see in-memory acceleration of these analytics stores for just such a purpose in the future.

The second item is far more powerful. The knowledge gained by processing big data should inform decisions. Moving that knowledge to the front of the data pipeline using an in-memory operational database enables decisions, driven by deep analytical understanding, to be operationalized and acted on each time an event enters the system.

Returning to the smart grid example, if our system is working as described up to this point, we are making operational decisions on smart meter and grid-based readings. We are using data from the current month to access trending of components, determine billing, and provide grid management. We are exporting that data back to big data systems where scientists can explore seasonality trends, informed by data gathered about certain events.

Let's say these exploratory analytics have discovered that, given current grid scale, if a heat wave of +10 degrees occurs during the late summer months, electricity will need to be diverted or augmented from other providers. This knowledge can now be used within our operational system so that if/when we get that +10-degree heat wave, the grid will dynamically adjust in a way that's both informed by history and based on current data. We have closed the loop on the data intelligence within the power grid.

Not every customer is looking to solve all requirements of fast data at once, but most points are included in the typical requirements. It's risky to gloss over these requirements; it is important not to make a tactical decision on the fast data component because a data architect thinks, "I only have to worry about ingesting right now." This is a surefire path to refactoring the architecture, far sooner than might otherwise be the case.

CHAPTER 6

Fast Data Applications (and Most of Them Are)

At this point, it is natural to ask questions: What are these data-intensive applications? Where do they exist? While this book has presented a number of detailed use case examples, there is no shortage of places where fast data applications are producing new value for users and businesses.

Few industries are immune from the pressures and opportunities that vast amounts of data represent. However, as famously stated by William Gibson, "The future is already here—it's just not very evenly distributed."[1]

The early-to-mid market adoption of data-intensive applications can be segmented into three broad categories, based on the industry's progression to an evolved data-driven strategy.

[1] William Gibson on NPR's *Fresh Air*, August 1, 1993. Also in "The Science in Science Fiction" on Talk of the Nation, NPR (30 November 1999, Timecode 11:55).

Industries That Have Historically Dealt with Fast Data Challenges in a Siloed Way

Examples: Capital markets, telco

These uses have existed for a number of years, but have been siloed within the organization and have required high-cost, specialized systems to support the functionality they provide.

A modern enterprise data architecture offers these users the ability to reduce the costs of delivering their services, but more importantly provides the ability to use the data already being captured, along with new, additional data sources, in a much more broad context that provides better, smarter interactions.

Example: Telco billing has been a batch process for a long time. This process was characterized by collecting call detail records, enriching those records in a batch process, and delivering a customer a consolidated bill at the end of a billing cycle. By combining fast data into the enterprise data architecture, telco providers are able to offer immediate, realtime services, such as Bill Shock notification; customized pricing plans; and realtime billing services.

Industries Being Transformed by the Changes Data Represents

Examples: Consumer web, mobile, gaming, advertising

These industries are well situated to take advantage of the increased power of data. The products and services they offer naturally create data based on the interaction their customers have with their products; the availability of that data represents opportunity.

The modern enterprise data architecture brings far greater customization, personalization, and associated benefits to customers. The use of this data in customer interactions creates improved customer experiences, enables the creation of more customized services, and provides an opportunity to increase profitable interactions.

Example: Online advertising has chased the same elusive goal all advertising has sought—delivering the right ad, to the right audience, at the right time. But early entrants into the digital advertising world were unable to get closer to that ideal than the print or broadcast advertisers they were attempting to replace. A generic ad on an automotive website was no more targeted than a generic ad in an automotive magazine. Now, with the addition of a data-driven architecture, the ad can be targeted based on demographic trends (historic), current user profile (static), and the previous clicks and current performance of the various advertising exchange options (real time).

Future Applications Where Data Is the Major Value

Examples: Industrial Internet, smart infrastructure, Internet of Things

Perhaps the most promising improvements to everyday life will come from areas just now emerging. These industries have not been automated to the extent that we will see them automate in the next five years. Data will be the driving factor in the value these services offer. Unlike the category above, these industries need to build the endpoints that will be controlled by the smart data they generate.

The enterprise data architecture will enable the intelligence of these industries.

> *A large measure of the utility of these services will come not from the devices themselves, but from the ancillary services and intelligence derived from the data.*

Example: Smart meter deployments initially look like a way to reduce the human labor involved in the process of reading a meter. But that is a small, and likely not even cost-effective, benefit of the smart electrical meter. The ability of the meter to communicate bi-directionally, to be considered in the context of the surrounding environment to warn of imminent disasters, maintenance needs, and efficiency advantages, are all high value–added services made possible by data.

Fast data applications will, of course, move beyond the use cases presented above as more traditional users—in Geoffrey Moore's terminology, the Pragmatists, Conservatives, and Luddites—feel pressure to extract value from business data to remain competitive. Imagine if the taxi industry, for example, had seen the value in data before Uber, Lyft, et al., emerged to disintermediate its business model. While one could argue that the nature of the traditional taxi industry does not lend itself to a broad sharing and analysis of data, it is clear that markets can be created—and destroyed—by the ability (or failure) to recognize fast data opportunities.

CHAPTER 7

How Fast and Big Applications Will Enter the Enterprise

Fast data is already streaming into the enterprise, and more is coming on a daily basis. However, in many cases, enterprises are pushing this fast data directly into the data lake, missing the opportunity to extract valuable realtime insights from data streams using in-memory technology. Realizing the benefits of this fast data requires a new enterprise data architecture. Therefore, the way in which systems are designed and built to leverage streams of data will define how quickly and pervasively fast data applications will be rolled out within an organization.

To understand how enterprise adoption of fast data technologies will occur, one needs to examine both the data sources and the applications that utilize those data sources. Four broad usage environments will drive enterprise adoption of fast data. The first three are combinations of a specific application and the data source(s) that encompass that application. The fourth category will be defined by corporations that truly understand the value that exists in being data-driven, and are prepared to implement an enterprise data architecture designed to unify all data interaction within the enterprise.

Existing Applications

This category of usage exists when applications that manage data begin to experience increasing volumes of data, exerting pressure on existing applications. Given the normal architecture of these systems, the load on the traditional database component will no longer meet the needs of the application; a change in the application will be required.

Adoption will occur because systems are no longer capable of meeting the needs of application users. Application developers will be forced to look at alternative technologies as the rate of inbound events exceeds what is possible to manage with the more traditional database systems around which applications were originally designed.

For example, this is what has happened with mobile subscriber data, and more change is coming. As phones and phone service prices continue to drop, more customers are coming online in a given geography; new markets are opening because of the lower-cost model. A subscriber system using a traditional database to manage one million subscribers will break under the load when demand expands to 100 million subscribers.

Equally taxing on the system is when a process that has historically had relatively few inputs is enhanced to add more detailed measurements. As an illustration, manufacturing Enterprise Resource Planning (ERP) software does not appear to be a likely candidate for fast data until one realizes that entire manufacturing lines are being retrofitted with sensors on every component in the manufacturing process. These systems are developed to feed realtime manufacturing data back into resource planning software to enable fine-grained adjustments and optimizations. These changes put enormous stress on systems, often forcing evaluation of new database technology.

New Applications, Existing Data Sources

Another way in which existing data sources are driving enterprise adoption of fast data is when those data sources, which have existed for years, are deemed to have newfound value. This newfound value often manifests itself in two ways: looking at data as it is generated in real time, or looking at it differently or in combination with other activities. Occasionally this triggers a displacement of one set of tools for either a broader or more customized solution. The change driven in these applications will remain within the confines of the single application, but will allow for more innovative uses by the application developer.

Consider an example: Network packets are not a new data source within the enterprise. However, fast data technologies have advanced to the point at which network packet ingestion creates new capabilities from already existing data. These network packets can be the source of fraud detection or Distributed Denial of Service (DDoS) detection by harnessing data currently available in the enterprise.

New Applications, New Data Sources

New data sources enable companies to launch new products and services, creating disruptive forces in many industries. These applications marry inbound data and user and device interaction with the environment to create new categories of products.

In many cases, these systems are being built as a complete package—for example, new smart infrastructure systems that manage power distribution in many cities. But some come from the ability to take sensor information from a phone and build entirely new applications on data that was not previously available. What makes these products notable is that a large portion of the value the user experiences with the product derives from the data that informs the product's interaction.

New Data-Driven Enterprise Integration

While the categories discussed previously are likely to be the initial adoption path into enterprise fast data, they are not the most disruptive. As reviewed in the beginning of this book, there is a 1+1=3 opportunity when all data assets within an enterprise are combined in an enterprise data architecture that can leverage all data—structured and unstructured, real time and historic, fast and big—across product lines and business units.

Companies that see the opportunity and move quickly to adopt an enterprise data architecture will gain the most from the imminent disruption that will come from capturing and utilizing both fast and big data.

CHAPTER 8

Getting There: Making the Right Fast Data Technology Choices

Application developers and technical managers involved with building fast and big data applications have a number of technology alternatives to evaluate. Clearly, the choices made in all phases of the architecture are important, but special attention must be paid to the choices for the fast data portion of the system.

Architectural Approaches to Delivering Fast Data

Three technology categories can be evaluated as the core components for the fast data portion of the enterprise data architecture: fast OLAP systems, stream processing products, and fast operational database systems. All are highly capable systems, but some are better suited to meet the broad requirements of fast data as described in this book. Organizing the alternatives by their core architecture types provides a way to evaluate strengths and weaknesses.

Fast OLAP Systems

New in-memory OLAP systems are able to drastically reduce reporting times and enable near realtime analysis of fast-arriving data. Many of these systems are column stores, optimized for uses where the only requirement is to improve reporting speeds. Additionally, some of these systems have the ability to ingest data quite quickly.

OLAP solutions, however, are designed as analytics engines and generally are not useful for making decisions on individual events as they arrive in the system. This inability to provide transactions at the point of data entering the architecture restricts these systems from solving the primary value that is achieved in the fast data portion of the architecture.

Stream Processing Systems

Stream processing approaches, including complex event processing (CEP), are available as open source as well as commercial options. Stream processing has been around for decades and has proven valuable in some very specialized uses in specific industries such as capital markets trading, where very specific patterns and timings need to be identified. When used in these environments, it is a well-suited system.

Stream processing systems provide scalable message processing and coordination between systems that often scales across commodity servers. However, stream processing systems do not maintain data state. As a result, they are severely limited in the ways in which they interact with an event entering the pipeline. All context of other data, either static data in data fusion instances, or changing data from other events passing through the system, is lost. Also, without the concept of state, analytics are performed by hand-coding algorithms and maintaining and managing the state for the results.

Because stream processing wasn't designed to serve the needs of modern fast data applications, it tends to be a poor match. In order to overcome these shortcomings, additional code is often written to perform continuous computations (realtime analytics), and databases are added to maintain state. This adds complexity and moves the performance bottleneck to another component in the system. The results are often systems that don't meet the requirements of the application and are burdened with complexity.

Operational Database Systems

Operational database systems are, by definition, designed to support per-event decision-making that is informed by other data stored within the system. Operational databases have long been the standard for interactive applications, but historically were unable to meet the performance required of fast data use cases.

In-memory, NewSQL systems are now available that are capable of meeting the performance requirements of the operational work as well as delivering full dataset analytics. Because these systems were designed with fast data applications in mind, the integration with the big data portion of the architecture is normally built in.

	Fast OLAP	Stream processing	Fast operational DB
Ingests data streams	Some	Yes	Yes
Data-driven event decisions	No	No	Yes
Realtime analytics	Yes	Through add-on	Yes
Integrates with big data system	No	Yes	Yes
Serves analytic results from big data systems	Yes	No	Yes

CHAPTER 9
Conclusion

Understanding the promise and value of fast data is an absolute necessity, but it is not sufficient to guarantee success for companies still working to implement big data initiatives. Having the tools, and the skills, to take advantage of fast data is critical for businesses in all industries and geographies.

Fast data is the payoff for big data. While much can be accomplished by mining data to derive insights that enable a business to grow and change, looking into the past provides only hints about the future. Simply collecting vast amounts of data for exploration and analysis will not prepare a business to act in real time, as data flows into the organization from millions of endpoints: sensors, mobile devices, connected systems, and the Internet of Things.

Because fast and big data have different requirements, it's necessary to have a component on the front end of the enterprise data architecture to ingest and interact on data, perform *real* realtime analytics, and make data-driven decisions on each event. Applications can take action, and data can be exported to the data warehouse for historical analytics, reporting, analysis, and more.

The missing link between fast and big is a unified enterprise data architecture. This approach links high-value, historical data from the data lake to fast-moving, inbound data from multiple endpoints. This frees application developers to write code that adds value to the organization, rather than being burdened by writing code to persist data as it flows to the data lake. An in-memory operational system that can decide, analyze, and serve results at fast data's speed is key to making big data work at enterprise scale.

Fast data, achieved through adoption of a new enterprise data architecture, gives organizations the tools to process high-volume streams of data while enabling millions of complex decisions in real time. With fast data, things that were not possible before become achievable: instant decisions can be made on realtime data to drive sales, connect with customers, inform business processes, and create value.

About the Author

Scott Jarr is a technology visionary who brings over 20 years of experience building, launching, and growing groundbreaking software companies.

In 2010, Scott cofounded VoltDB after realizing the opportunities available to businesses that could effectively use data to impact the world. Prior to VoltDB, Scott founded, served as board member, and advised several early-stage companies in the data, mobile, and storage markets. As a key member of the executive team at SaaS pioneer LiveVault, he was instrumental in growing the business, leading to its successful acquisition. Scott has an undergraduate degree in mathematical programming from the University of Tampa and an MBA in entrepreneurship from the University of South Florida.

As part of his commitment to fostering the entrepreneurial spirit in others, Scott serves as a board member and advisor helping other early-stage companies build their businesses.

Colophon

The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.

Building Data Science Teams
DJ Patil
What is Data Science?
Mike Loukides
Data Jujitsu
The Art of Turning Data Int
DJ Patil
Planning for Big Data
A CIO's handbook to the changing data landscape
O'Reilly Radar Team
O'REILLY
SLEEK & SMART
Big Dat
When you don't have time to ...around
Making Data Work