Featured in Development

Big Data Analytics with Spark Book Review and Interview

Big Data Analytics with Spark book, authored by Mohammed Guller, provides a practical guide for learning Apache Spark framework for different types of big-data analytics projects, including batch, interactive, graph, and stream data analysis as well as machine learning. InfoQ spoke with author about the book & development tools for big data applications.

All in Development

- Architecture & Design
  - Architecture
  - Enterprise Architecture
  - Scalability/Performance
  - Design
  - Case Studies
  - Microservices
  - Patterns
  - Security

Featured in Architecture & Design

Big Data Analytics with Spark Book Review and Interview

Big Data Analytics with Spark book, authored by Mohammed Guller, provides a practical guide for learning Apache Spark framework for different types of big-data analytics projects, including batch, interactive, graph, and stream data analysis as well as machine learning. InfoQ spoke with author about the book & development tools for big data applications.

All in Architecture & Design

- Data Science
  - Big Data
  - Machine Learning
  - NoSQL
  - Database
  - Data Analytics
  - Streaming

Featured in Data Science

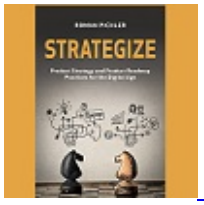
Big Data Analytics with Spark Book Review and Interview

Big Data Analytics with Spark book, authored by Mohammed Guller, provides a practical guide for learning Apache Spark framework for different types of big-data analytics projects, including batch, interactive, graph, and stream data analysis as well as machine learning. InfoQ spoke with author about the book & development tools for big data applications.

All in Data Science
- Culture & Methods
  - Agile
  - Leadership
  - Team Collaboration
  - Testing
  - Project Management
  - UX
  - Scrum
  - Lean/Kanban
  - Personal Growth

Featured in Culture & Methods

Q&A with Roman Pichler about Strategize



The book Strategize by Roman Pichler provides practices, advice, and examples for product strategy and roadmapping that you can use to create successful products. InfoQ interviewed Pichler about applying product strategy and roadmapping with agile, innovation in product strategy, eliminating features when defining

products, different kinds of roadmaps, and on measurements for product management.

All in Culture & Methods

- DevOps
  - Infrastructure
  - Continuous Delivery
  - Automation
  - Containers
  - Cloud

Featured in DevOps

Virtual Panel on (Cloud) Lock-In


There's no shortage of opinions on the topic of technology lock-in. InfoQ reached out to four software industry leaders to participate in a lively virtual panel on this topic: Joe Beda, Simon Crosby, Krish Subramanian, and Cloud Opinion.

All in DevOps

San Francisco Nov 7-11

London Mar 6-10, 2017

New York Jun 26-30, 2017

- Mobile
- HTML5
- JavaScript
- APM
- Cloud
- Java
- Microservices
- Big Data
- Database

All topics

Interview

Big Data Analytics with Spark Book Review and Interview

Posted by Srini Penchikala on Jun 23, 2016 | Discuss

- Share
- "Read later"
- "My Reading List"

Key takeaways

- Learn how to use Apache Spark for different types of big-data analytics use cases like batch, interactive, graph, streaming data and machine learning.
- Understand the Spark Core and its add-on libraries, Spark SQL, Spark Streaming, GraphX, MLlib, and Spark ML.
- Learn about development and testing tools for developers to use when working on projects using Spark.
- Best practices for the performance and tuning of Spark programs.
- Read about how Spark works with cluster setup, management and monitoring.

Big Data Analytics with Spark book, authored by Mohammed Guller, provides a practical guide for learning Apache Spark framework for different types of big-data analytics projects, including batch, interactive, graph, and stream data analysis as well as machine learning. It covers Spark core and its add-on libraries, including Spark SQL, Spark Streaming, GraphX, MLlib, and Spark ML.

Readers can learn how to perform data analysis using in-memory caching and advanced execution engine components of Apache Spark framework.

Author talks about how to use Spark as a unified platform for data processing tasks like ETL pipelines, business intelligence, real-time data stream processing, graph analytics, and machine learning. He also discussed other topics like cluster managers and monitoring of Spark programs.

The book includes an introduction to other technologies and frameworks that are commonly used with Spark, such as distributed file management systems (HDFS), Avro, Parquet, distributed messaging (Kafka), NoSQL databases (Cassandra, HBase), and cluster management (Mesos).

InfoQ spoke with Mohammed Guller about the book, Spark framework and tools for developers who are working on big data applications using Spark.

InfoQ: How do you define Apache Spark framework and how it helps with big data analytics projects and initiatives?

> Guller: Apache Spark is a fast easy-to-use general-purpose cluster computing framework for processing large datasets. It gives you both scale and speed. More importantly, it has made it easy to perform a variety of data processing tasks on large datasets. It provides an integrated set of libraries for batch processing, ad hoc analysis, machine learning, stream processing and graph analytics.
>
> Data is growing almost exponentially. In addition, most of the data generated today is not structured, but either multi-structured or unstructured. Traditional tools such as relational databases cannot handle the volume, velocity or the variety of the data generated today. That is why you need frameworks such as Spark. It makes it easy to handle the three V's of big data. Another important thing to keep in mind is that organizations process or analyze data in different ways to get value out of it. Spark provides a single platform for different types of data processing and analytical tasks. You don't need to duplicate code or data, unlike special-purpose frameworks that do either only batch or stream processing.

InfoQ: Can you discuss what all development and testing tools are available for developers to use when they are working on projects using Spark?

> Guller: In general, developers can use whatever tools are available for the programming languages supported by Spark. Currently, Spark supports Scala, Java, Python and R.
>
> Let's take Scala as an example. Spark comes pre-packaged with an interactive development environment known as spark-shell. It is based on the Scala REPL (Read Evaluate Print Loop) tool. It provides a quick and easy way to get started with Spark. In addition, developers can use standard Scala IDEs such as Eclipse and IntelliJ IDEA. And if you are not a big fan of IDEs, you can write your code in your favorite text editor and compile it with SBT (Simple Build Tool).

InfoQ: What are some best practices you suggest to new developers who are just learning to use Spark framework?

> Guller: The best way to learn Spark is to experiment with it and write code using the Spark API. The concepts become much clear when you write and execute code. This is generally true for learning any new language or tool.
>
> Even though Spark is a big data processing framework, you don't necessarily need to have a big cluster or large datasets to learn Spark. You can run Spark on your laptop with a small dataset and get comfortable with the API and various libraries provided by Spark. My book has a chapter on how to get easily started with Spark.

InfoQ: How do you compare the different programming languages Spark currently supports, Scala, Java, Python, and R? What do you recommend for new developers if they have to choose a language, which one should be?

> Guller: Spark itself is written in Scala. So, historically, Scala was a first class citizen and support for other languages lagged behind a little bit. However, that gap is shrinking with every new release of Spark. Similarly, Spark applications written in Scala had speed advantage over programs written in Python applications. However, with all the new optimizations provided by Spark under the hood, the speed difference has also reduced.

I personally like Scala, because it increases your productivity and enables you to write concise and better quality code. It rekindled my love for programming.

Having said that, a developer can use whatever language they are most comfortable with. So if you are a Python guru, use Python. You don't need to switch or learn a new language as long as you know one of the languages supported by Spark.

If you want to learn a new language and get optimal performance, I recommend Scala. That is the reason I included a chapter on functional programming and Scala in my book.

InfoQ: What is the best way for cluster setup of Spark on the local machine or on the cloud?

Guller: Spark provides a script, spark-ec2, for setting up a Spark cluster on Amazon AWS. This script allows you to launch, manage and shut down a Spark cluster on Amazon cloud. It installs both Spark and HDFS. It is a pretty flexible script with a number of input arguments, allowing you to create custom clusters for your specific processing needs and budget.

InfoQ: Can you talk about the real-time streaming data analytics using Spark Streaming library?

Guller: The Spark Streaming library extends Spark for stream processing. It provides operators for analysing stream data in near real-time. It uses a micro-batching architecture. Essentially, a stream of data is split into micro-batches. The batch interval is specified by a developer. Each micro-batch is represented by an RDD (Resilient Distributed Dataset), which is Spark's primary data abstraction.

The micro-batching architecture has both advantages and disadvantages. On the plus side, it provides high-throughput. So Spark Streaming is great for performing analytics on stream data. However, if your application needs to process each event in a stream individually with very low-latency (milliseconds) requirements, Spark Streaming may not be a good fit.

InfoQ: What are the considerations for performance and tuning of Spark programs?

Guller: This is a vast topic, since Spark provides many knobs for performance tuning. I will discuss some of the important things to keep in mind.

First, for most data processing applications, disk I/O is a big contributor to application execution time. Since Spark allows you to cache data in memory, take advantage of that capability whenever you can. Caching data in memory can speed up your application by up to 100 times. Obviously, this means it is better to setup your Spark cluster with machines with large amount of memory.

Second, avoid operators that require data shuffling. Shuffling data across a network is an expensive operation. Keep this in mind when you write your data processing logic. Sometimes the same logic can be implemented with a more efficient operator. For example, instead of the groupByKey operator, use the reduceByKey operator.

Third, optimize the number of partitions in your data. If your data is not correctly partitioned, you are not taking advantage of the data parallelism provided by Spark. For example, assume you have a Spark cluster with 100 cores. But if your data has only 2 partitions, you are underutilizing your compute power.

Fourth, co-locate data node with compute node for optimal performance. For example, if your data is in HDFS, install Spark on the same HDFS cluster. Spark

will execute data processing code as close to data as possible. For example, it will first try to execute a task on the same machine where data is located. If cannot execute a task on that machine, it will try to find a machine on the same rack. If that is not possible, it will use any machine. Minimize both disk and network I/O.

These are some of the common performance related things to keep in mind.

InfoQ: What is the current support for securing Spark programs so only authorized users or applications will be able to execute those programs?

Guller: Spark supports two methods of authentication: shared secret and Kerberos. The shared secret authentication mechanism can be used with all the cluster managers: YARN, Mesos and Standalone. In addition, YARN allows you to use Kerberos with Spark.

Spark also supports encryption using SSL and SASL. SSL is used for securing communication protocols, while SASL is used for securing block transfer service.

InfoQ: How do you monitor Spark programs using Spark Web Console and other tools? What metrics do you typically measure in Spark programs?

Guller: Spark provides comprehensive monitoring capabilities. My book has a complete chapter on this topic. Spark not only exposes a wealth of metrics, but also provides web-based UI for monitoring both Spark cluster and applications running on it. In addition, it supports third-party monitoring tools such as Graphite, Ganglia and JMX-based monitoring applications.

I use monitoring for both performance optimization and debugging. The specific metrics that I review depends on the problem that I am trying to solve. For example, you can use the monitoring UI to check the state of your cluster and allocation of resources amongst your applications. Similarly, you can use the monitoring UI to see the amount of parallelism within the jobs submitted by your application. You can also check the amount of data processed by different tasks and the time taken. It can help you find straggler tasks. These are just a few examples.

InfoQ: What are some new features you would like to see added in the future releases of Spark framework?

Guller: The Spark developer community has done an amazing job in enhancing Spark with every new release. So my wish list is not big. Most of the new features that I would like to see are related to machine learning.

One thing that I feel is missing from Spark is a graphing or data plotting library for Scala developers. Exploratory visualization is a critical part of data analysis. R developers can use ggplot2. Python has matplotlib. It would be good to have something similar for Scala developers.

Another thing that I would like to see is Spark's statistical and machine learning library become on par with that provided by R. Finally, I would like to see better support for being able to export and import machine learning models using standards such as PMML and PFA.

InfoQ: Spark Machine Learning currently provides several different algorithms. Do you see any other ML libraries that may add value to machine learning & data science needs of the organizations?

Guller: You are right in that Spark's machine learning library comes with a rich set of algorithms. In addition, new algorithms are added in every release.

Spark can be used with external machine libraries, so whatever capabilities Spark is missing, those gaps can be filled with other libraries. For example, Stanford CoreNLP library can be used with Spark for NLP-heavy machine learning tasks. Similarly, SparkNet, CaffeOnSpark, DeepLearning4J, or TensorFlow can be used with Spark for deep learning.

Guller talked about the value Spark framework brings to the table.

Guller: Spark is a great framework for analyzing and processing big data. It is easy to use and provides a rich set of libraries for a variety of tasks. Plus, it provides scale and speed for processing really large dataset. Anyone working with Big Data or eager to get into the Big Data space should definitely learn it.

He also said he gets questions from lot of people about the relationship between Hadoop and Spark. He responded to the following two questions he gets from time to time.

InfoQ: Will Spark Replace Hadoop?

Guller: The short answer is no. Today Hadoop represents an ecosystem of products. Spark is a part of that ecosystem. Even core Hadoop consists of three components: a cluster manager, a distributed compute framework and a distributed file system. YARN is the cluster manager. MapReduce is the compute framework and HDFS is the distributed file system. Spark is a successor to the MapReduce component of Hadoop.

Many people are either replacing existing MapReduce job with Spark jobs or writing new jobs in Spark. So you can say that Spark is replacing MapReduce, but not Hadoop.

Another important thing to keep in mind is that Spark can be used with Hadoop, but it can also be used without Hadoop. For example, you can Mesos or the Standalone cluster manager instead of YARN. Similarly, you can use S3 or other data sources instead of HDFS. So you don't need to install Hadoop to use Spark.

InfoQ: Why are people replacing MapReduce with Spark?

Guller: Spark offers many advantages over MapReduce.

First, Spark is much faster than MapReduce. Depending on the application, it can be up to 100 times faster than MapReduce. One reason Spark is fast because of its advanced job execution engine. Spark jobs can have any number of stages, unlike MapReduce jobs which always have two stages. In addition, Spark allows applications to cache data in memory. Caching tremendously improves application execution time. Disk I/O is a significant contributor to application execution time for data processing application. Spark allows you to minimize Disk I/O.

Second, Spark is easy to use. Spark provides a rich expressive API with 80+ operators, whereas MapReduce provides only two operators Map and Reduce. The Spark API is available in four languages, Scala, Python, Java and R. You can write the same data processing job in Scala/Spark using 5x-10x less code than the amount of code you will have to write in MapReduce. Thus Spark also significantly improves developer productivity.

Third, Spark offers a single tool kit for a variety of data processing task. It comes prepackaged with integrated libraries for doing batch processing, interactive analytics, machine learning, stream processing and graph analytics. So you don't need to learn multiple tools. In addition, you don't need to duplicate code and data at multiple places. Operationally also, it is easier to manage one cluster instead of multiple special-purpose clusters for different types of jobs.

About the Book Author

Mohammed Guller is the principal architect at Glassbeam, where he leads the development of advanced and predictive analytics products. Over the last 20 years, Mohammed has successfully led the development of several innovative technology products from concept to release. Prior to joining Glassbeam, he was the founder of TrustRecs.com, which he started after working at IBM for five years. Before IBM, he worked in a number of hi-tech start-ups, leading new product development. Mohammed has a master's of business administration from the University of California, Berkeley, and a master's of computer applications from RCC, Gujarat University, India.

- Personas
- Data Science
- Architecture & Design
- Development
- Topics
- Database
- Apache Spark
- Infrastructure
- Data Analytics
- InfoQ
- Data Analysis
- Book Review
- Big Data

Hello stranger!

You need to Register an InfoQ account or Login or login to post comments. But there's so much more behind being registered.

Get the most out of the InfoQ experience.

Tell us what you think

Please enter a subject

Message

Allowed html: a,b,br,blockquote,i,li,pre,u,ul,p

Email me replies to any of my messages in this thread

Post Message

Community comments Watch Thread

Close

by

on

- View
- Reply
- Back to top

Close

Subject Your Reply

Quote original message

Allowed html: a,b,br,blockquote,i,li,pre,u,ul,p

Email me replies to any of my messages in this thread

Post Message Cancel

Close

Subject Your Reply

Allowed html: a,b,br,blockquote,i,li,pre,u,ul,p

Email me replies to any of my messages in this thread

Cancel

Close

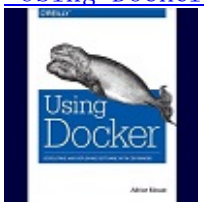
OK

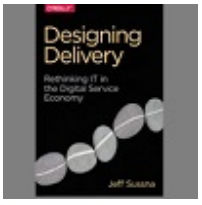
RELATED CONTENT

- Predicting the Future: Surprising Revelations from Truly Big Data May 25, 2016

- Big Data Processing with Apache Spark - Part 4: Spark Machine Learning May 15, 2016

- Big-Data Analytics Misconceptions May 04, 2016
- "Using Docker" Book Review and Q&A with Author Adrian Mouat Apr 02, 2016





- Designing Delivery Book Review and Interview Mar 30, 2016

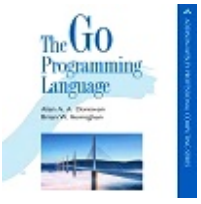
- Spark in Action Book Review & Interview Mar 18, 2016
- Big Data Processing with Apache Spark - Part 3: Spark Streaming Jan 07, 2016

- Getting Ready for IoT’s Big Data Challenges with Couchbase Mobile Jan 20, 2016

- Book Review: All About Java 8 Lambdas Mar 08, 2016

- Apache Spark for Big Data Processing Feb 15, 2016

- Book Review: The Go Programming Language Feb 06, 2016



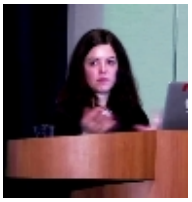
been asked if PostgreSQL could replace MongoDB as a JSON database. There's a short answer to that, but we'd prefer to show you. Learn more.

- Mongoosastic: The Power of MongoDB & Elasticsearch Together

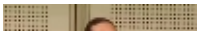
  Learn how to use MongoDB and Elasticsearch at the same time without rebuilding your code.

RELATED CONTENT

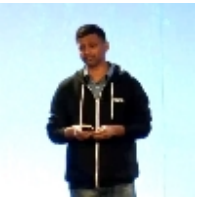
- Machine Learning with Spark: Book Review and Interview Jan 15, 2016
- "Elasticsearch in Action" - Book Review and Authors Interview Jan 13, 2016

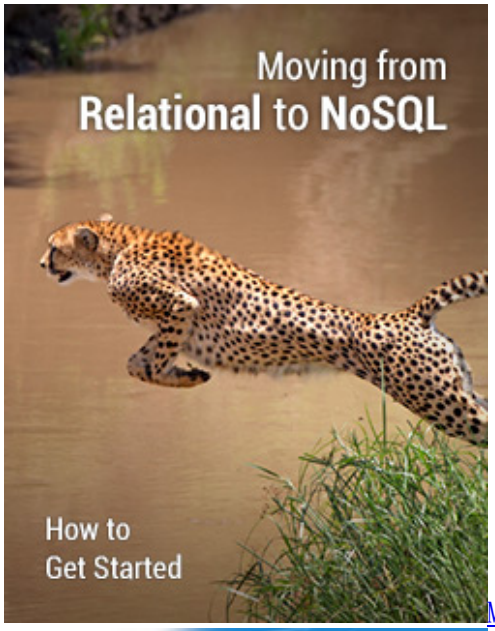


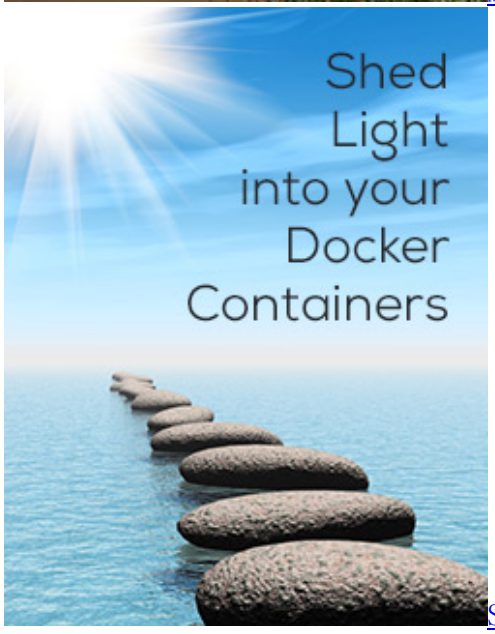
- Rethinking Streaming Analytics for Scale Apr 04, 2016
- Neha Narkhede: Large-Scale Stream Processing with Apache Kafka Jun 20, 2016
- Apache Spark 2.0 Technical Preview Jun 01, 2016
- Data Streaming Architecture with Apache Flink Jun 10, 2016



- Hypermedia Web API as a Network of Data Jun 15, 2016 
- The Mechanics of Testing Large Data Pipelines Apr 25, 2016 
- Understanding Real-time Conversations on Facebook Apr 18, 2016 
- The Role of a Data Scientist in 2016 Mar 27, 2016 

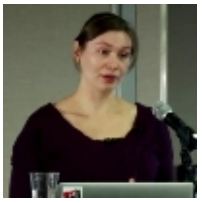
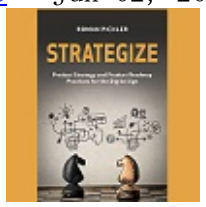
- Real-time Stream Computing & Analytics @Uber Apr 10, 2016 

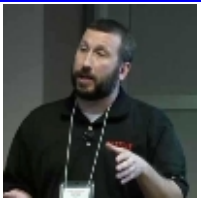
- 

Moving from Relational to NoSQL: How to Get Started
- 

Shed Light into your Docker Containers

RELATED CONTENT

- Insights from History of Rock Music via Machine Learning Mar 23, 2016 
- Supercharging Operations and Analytics: Using Spring XD to Support Analytics and CEP Feb 29, 2016 
- Lana Gibson on Using Analytics to Influence Content Design Feb 28, 2016 
- IoT Realized - The Connected Car v2 Jan 31, 2016 
- The Lego Model for Machine Learning Pipelines Jan 17, 2016 
- LinkedIn Details Production Kafka Debugging and Best Practices Jun 16, 2016
- LinkedIn Details Open-Sourced Kafka Monitor Jun 08, 2016
- Confluent Platform 3.0 Supports Kafka Streams for Real-Time Data Processing Jun 04, 2016
- Cloudera Announces Partnership with the Broad Institute Jun 02, 2016
- Q&A with Roman Pichler about Strategize Jun 21, 2016 

- Netflix Keystone - How We Built a 700B/day Stream Processing Cloud Platform in a Year May 20, 2016 

InfoQ Weekly Newsletter

Subscribe to our Weekly email newsletter to follow all new content on InfoQ

Your email here Subscribe

Development

Big Data Analytics with Spark Book Review and Interview

Safari 10 Ships WebDriver

C# 7 and Beyond with Mads Torgersen

Architecture & Design

Big Data Analytics with Spark Book Review and Interview

Cloud Identity Summit Pushes Change in Identity and Security

C# 7 and Beyond with Mads Torgersen

Culture & Methods

Why Diversity and Inclusion Matters, and How to Drive It

Q&A with Roman Pichler about Strategize

Kevin Miyashiro on Agilility Readiness Canvas

Data Science

Big Data Analytics with Spark Book Review and Interview

Apache TinkerPop Graduates to Top-Level Project

Test Well and Prosper: The Great Java Unit-Testing Frameworks Debate

DevOps

Cloud Identity Summit Pushes Change in Identity and Security

Five Ways to Not Mess Up Microservices in Production

Virtual Panel on (Cloud) Lock-In

InfoQ Weekly Newsletter

Subscribe to our Weekly email newsletter to follow all new content on InfoQ