Big Data Software Analytics with Apache Spark

Georgios Gousios
Delft University of Technology
Delft, The Netherlands
g.gousios@tudelft.nl

ABSTRACT

At the beginning of every research effort, researchers in empirical software engineering have to go through the processes of extracting data from raw data sources and transforming them to what their tools expect as inputs. This step is time consuming and error prone, while the produced artifacts (code, intermediate datasets) are usually not of scientific value. In the recent years, Apache Spark has emerged as a solid foundation for data science and has taken the big data analytics domain by storm. We believe that the primitives exposed by Apache Spark can help software engineering researchers create and share reproducible, high-performance data analysis pipelines.

In our technical briefing, we discuss how researchers can profit from Apache Spark, through a hands-on case study.

CCS CONCEPTS

•Software and its engineering → Software libraries and repositories; *Software configuration management and version control systems; Collaboration in software development;*

KEYWORDS

data analytics; big data; Apache Spark

ACM Reference format:
Georgios Gousios. 2018. Big Data Software Analytics with Apache Spark. In *Proceedings of 40th International Conference on Software Engineering , Gothenburg, Sweden, May 27-June 3, 2018 (ICSE '18 Companion),* 2 pages.
DOI: 10.1145/3183440.3183458

1 DESCRIPTION

The availability of Open Source Software, apart from the software landscape, also changed how research in software engineering is being done. Researchers could, for the first time, get their hands on rich data sources such as version control system repositories and bug databases *en masse*. This trend was only amplified when platforms such as GitHub integrated not only software repositories but also issue databases, wikis and collaboration mechanisms for millions of repositories. Based on them, datasets, such as GHTorrent [4], made terabytes of data readily available for analysis. Consequently, quantitative empirical software engineering is increasingly becoming a data science discipline [8].

A common pattern in data science projects is the *ETL cycle*: researchers need to **E**xtract data from raw sources, **T**ransform them in a format suitable for analysis and **L**oad them in analysis environments, such as R and SciPy. To automate this process and to ensure reproducibility of results, data scientists need to construct data analysis *pipelines*, where a series of custom made scripts glue together existing or purpose-made tools. This is also the prevailing modus operandi in quantitative software engineering and software analytics domains.

Apache Spark is a distributed computing platform, specifically tuned for interactive real time data analysis. Effectively, Spark hides the details of distributed processing behind a convenient API. With Spark, researchers can map their data sources into immutable lists or data frames and transform them using a declarative API based on functional programming primitives (higher-order functions such as `map`, `fold` and `groupBy`). In the background, Spark will partition the original data, distribute the partitions to a cluster of machines, optimize the user-provided computations in a way that it minimizes data movement and apply them in parallel. Due to its convenient API and raw execution speed, Spark has taken the big data world by storm, since its introduction in 2010 [11].

The data processing primitives exposed by Spark are extremely powerful and a natural fit for common types of data-driven software engineering research. As such, it can facilitate research in the fields of repository mining, software analytics, requirements engineering and software testing. Below, we present a few examples of how Spark can help with common tasks software engineering researchers perform:

Ad-hoc queries Spark can map Comma-Seperated Values (CSV) files to in-memory DataFrames, which researchers can then query with SQL without importing them in a database. Spark will run the queries in parallel on a cluster of machines.

Ad-hoc dataset joining Spark can connect, through JDBC, to all relational databases and most non-relational ones. This enables efficient join operations between data in arbitrary formats, for example between a CSV file and records in a MongoDB database.

Tool integration Spark can parallelize the execution of any user defined function as long as it does not have side effects. Static analysis tools written in Java can be adapted to work as part of a Spark `map` step and then be applied on multiple files or multiple software repositories in parallel.

Data conversion Processing textual raw data into a structured format (the extract phase in the ETL cycle) is a matter

ICSE '18 Companion, Gothenburg, Sweden

978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/3183440.3183458

of defining the end format and a function that converts the raw input to the required format. Spark can use almost any Java or Python library (e.g. JGit to interface with a Git repository) at this stage to parse raw data formats.

Machine learning Spark features extensive machine learning and graph processing libraries, along with comprehensive statistics functionality. For any usage not covered in the main libraries (e.g. plotting), researchers can tap on any library available in the Python or R ecosystems, as Spark features native Python and R front-ends, respectively.

While Apache Spark is now an industry standard, there are very few works (e.g., [10]) that use it in quantitative software engineering. In our view, this leads to wasted hours on scaling custom data processing tools and debugging mistakes that have been solved already. For this reason, in our technical briefing, we will focus on the essentials of applying Apache Spark on software engineering data, in an attempt to make software engineering studies better. We will be covering the following topics:

- *Review functional programming basics:* This is necessary, as most (all?) current big and streaming data analysis tools are based on functional programming primitives.
- *Present Apache Spark in a nutshell:* What are RDDs and what are Dataframes? How can we use SparkSQL for analysing tabular data? How does Spark distributed the processing on computer clusters and what affects execution performance?
- *Present a live demo of applying Apache Spark on a software engineering task:*. How can we extract data from Git? How can we connect our pipeline to external data sources (e.g. GHTorrent?). How can we organize our experiments? How can we use Spark's machine learning library for a simple prediction task?

2 THE SPEAKER

The speaker has been practicing (and lately, teaching) big software data analysis since at least 2007, when he led the development of the Alitheia Core repository mining platform [6]. At the time, Alitheia Core could process 1-2 orders of magnitude more data than any comparable tool, custom or general. Later, he created GHTorrent [4], a custom real-time data collection infrastructure for data from the GitHub API, offering an accessible dataset to hundreds of repository mining researchers. He also contributed to several large scale data analysis pipelines, for analysing the whole Maven ecosystem [9], Travis CI builds [2] (also, co-created the TravisTorrent dataset [3]), developer analytics [1] and the whole JavaScript, Ruby and Rust ecosystems [7]. A short bio note follows:

Georgios Gousios is an assistant professor at the Software Engineering group, Delft University of Technology. His research interests include software engineering, software analytics and programming languages. He works in the fields of distributed software development processes, software quality, software testing, dependency management and research infrastructures. His research has been published in top venues, where he has received four best paper awards and various nominations. In total, he has published more that 60 papers and also co-edited the "Beautiful Architectures" book (OReilly, 2009). He is the main author of the GHTorrent data collection and curation framework and the Alitheia Core repository mining platform. Georgios holds and MSc from the university of Manchester and a PhD from the Athens University of Economics and Business, both in software engineering. In addition to research, he is also active as a speaker, both in research and practitioner-oriented conferences.

3 RELATED CONTENT

The content for this tutorial comes from the author's Big Data Processing course, given to 2nd year BSc students at TU Delft. To get acquainted with the theory behind the systems used for this tutorial, the participants are invited to study the course material [5].

REFERENCES

[1] Moritz Beller, Georgios Gousios, Annibale Panichella, and Andy Zaidman. 2015. When, How, and Why Developers (Do Not) Test in Their IDEs. In *Proceedings of the 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015)*. ACM, New York, NY, USA, 179–190. DOI: http://dx.doi.org/10.1145/2786805.2786843

[2] Moritz Beller, Georgios Gousios, and Andy Zaidman. 2017. Oops, My Tests Broke the Build: An Explorative Analysis of Travis CI with GitHub. In *Proceedings of the 14th Working Conference on Mining Software Repositories (MSR'17)*. IEEE press, 356–367. DOI: http://dx.doi.org/10.1109/MSR.2017.62

[3] Moritz Beller, Georgios Gousios, and Andy Zaidman. 2017. TravisTorrent: Synthesizing Travis CI and GitHub for Full-Stack Research on Continuous Integration. In *Proceedings of the 14th Working Conference on Mining Software Repositories (MSR'17)*. IEEE press, 447–450. DOI: http://dx.doi.org/10.1109/MSR.2017.24

[4] Georgios Gousios. 2013. The GHTorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13)*. 233–236. Best data showcase paper award.

[5] Georgios Gousios. 2018. Big Data Processing. (2018). http://gousios.org/courses/bigdata/book/preface.html

[6] Georgios Gousios and Diomidis Spinellis. 2009. A Platform for Software Engineering Research. In *MSR '09: Proceedings of the 6th Working Conference on Mining Software Repositories*, Michael W. Godfrey and Jim Whitehead (Eds.). IEEE, 31–40. DOI: http://dx.doi.org/10.1109/MSR.2009.5069478

[7] Riivo Kikas, Georgios Gousios, Marlon Dumas, and Dietmar Pfahl. 2017. Structure and Evolution of Package Dependency Networks. In *Proceedings of the 14th Working Conference on Mining Software Repositories (MSR '17)*. IEEE press, 102–112. DOI: http://dx.doi.org/10.1109/MSR.2017.55

[8] Tim Menzies, Laurie Williams, and Thomas Zimmermann. 2016. *Perspectives on Data Science for Software Engineering*. Morgan Kaufmann.

[9] Dimitris Mitropoulos, Vassilios Karakoidas, Panos Louridas, Georgios Gousios, and Diomidis Spinellis. 2014. The Bug Catalog of the Maven Ecosystem. In *MSR '14: Proceedings of the 2014 International Working Conference on Mining Software Repositories*. ACM, 372–365. DOI: http://dx.doi.org/10.1145/2597073.2597123

[10] Fabian Trautsch, Steffen Herbold, Philip Makedonski, and Jens Grabowski. 2017. Addressing problems with replicability and validity of repository mining studies through a smart data platform. *Empirical Software Engineering* (08 Aug 2017). DOI: http://dx.doi.org/10.1007/s10664-017-9537-x

[11] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster computing with working sets. *HotCloud* 10, 10-10 (2010), 95.