Computing (2020) 102:1409–1417
https://doi.org/10.1007/s00607-020-00822-9

EDITORIAL

Software-driven big data analytics

Guest editors' introduction

Rajiv Ranjan[1] · Zheng Li[2] · Massimo Villari[3] · Yan Liu[4] · Dimitrios Georgeakopoulos[5]

Published online: 5 June 2020

Abstract
Data analytics is the crucial step to reveal essential values of datasets and complete the value chain of big data. In practice, both the hardware infrastructure and the software stack play a fundamental role in big data analytics (BDA). Unfortunately, it is evident that a disproportionately larger amount of effort is being invested in the hardware infrastructure development over the software stack development. Given our concern about a software crisis brewing in the big data ecosystem, we argue that it is time to further strengthen and expand the role of software in BDA implementations. This special issue is then aimed to create a common ground and a reference point for both researchers and practitioners from multiple disciplines to discuss the rigor, relevance, experience and challenges of software-driven BDA as an emerging domain. We also expect to use this special issue to attract more attention and efforts to tighten communication and collaboration between the software engineering community and the data science community.

Keywords Big data analytics · Data science · Software · Software engineering

1 Introduction

Software is defining everything and dominating the world [35]. Accordingly, the booming big data ecosystem should also be part of this software-driven world. In particular, we believe that software rather than hardware guides the evolution direction of data analytical practices and research.

This work was supported in part by Chilean National Commission for Scientific and Technological Research (CONICYT, Chile) under Grant FONDECYT Iniciación 11180905.

Extended author information available on the last page of the article

1.1 Software-driven world

For more than 5 decades since the first software engineering conference in 1968 held in Garmisch, Germany, software has become increasingly pervasive across all the fields and has started ruling the world [25]. From the civilization's perspective, software stays at the center of the intelligence evolution and defines the future more than any other discipline [10]. In fact, substantial software systems have been proved able to and will continue to improve the sustainability of our society and promote the prosperity of humanity [8,23]. For example, in addition to the green application scenarios (e.g., paperless office software saves office cost and reduces carbon footprint), dedicated software systems can be employed to facilitate identifying, analyzing and optimizing the leverage points of multiple sustainability dimensions.

From an individual's perspective, the everyday life is "now built on software, without which life would be unimaginable" [9]. In particular, every aspect of our lives is experiencing digitalization to some degree. A typical sign is that we are surrounded by smarter and smarter devices (e.g., self-driving vehicles, smart phones, and smart watches) and environments (e.g., smart home, smart office, and smart city). Although the hardware components are widely recognized and possibly overemphasized even via the names [7], it is software that essentially makes those devices and environments smart. Inspired by the well-known metaphor of "mind versus brain" [26], after all, people have to rely on software systems to communicate and interact with the hardware objects around them.

From an organizational perspective, digital transformation is now an imperative movement happening in various organization bodies ranging from government agencies to industrial plants, while software is a critical and innovation-enabling component in the ongoing revolution of digital transformation [1]. Take industry as an example, there has been a wide consensus on (and almost a cliché claim) that every company will become a software company [30]. According to a recent survey in the traditional manufacturing areas [2], for instance, about half of a new vehicle's cost is determined by its electronics and software content, while a simple infusion pump may contain approximately 170,000 lines of code. More importantly, "the success in the industrial sector where data and communication equate to lost lives and billions of dollars largely depends on software's ability to create valuable functionality" [4].

1.2 Software-driven big data analytics

The emerging age of big data is leading us to an innovative way of understanding our world and making decisions. In particular, it is the data analytics that eventually reveals the potential values of datasets and completes the value chain of big data [19]. To obtain analytical results, there are naturally development and deployment requirements of appropriate functionalities, libraries, tools, systems, and software frameworks and solutions. Correspondingly, big data analytics (BDA) has become a new and crucial domain within the software-driven world.

The driver role played by software in data analytics can be traced back to "software-driven instrumentation" in 1985 [27]. Although by that time the goal was merely to

make an observation of some phenomenon of interest under controlled conditions, it had been realized that such instrumentation could help analysts make experimental and analytical measurements otherwise impossible or extremely expensive to make.

When it comes to implementing BDA, there are inevitably more challenges than traditional data analytical scenarios. On one hand, big data itself can cause significant performance problems in application programs in general, especially when involving databases [15]. On the other hand, following the No-Free-Lunch theorem [24], various data types and analytical demands might require completely different BDA applications involving different time and space complexities [6]. For example, de facto BDA workload characteristics tremendously vary, and the typical ones include batch-processing for offline analytical jobs, streamprocessing for real-time processing of data, query-processing with transactional features, and even a combination of them [16].

Given the aforementioned software nature of BDA applications, software engineering can act as a key to addressing the existing challenges in the BDA domain and to supporting different areas and aspects of BDA practices. It is even claimed that BDA has little to do with analytics but with software engineering [24]. From the software developer's perspective, the theories, processes, and techniques of software engineering can be introduced to the realization of efficient analytical operations [22]. From the software consumer's perspective, easy/ready-to-use platforms and facilities for satisfying BDA demands are urgent needs to serve data scientists who do not necessarily have expertise in software engineering (e.g., the Ophidia project [11]). In fact, both of these viewpoints have been highlighted by the European Commission as the future trends and research priorities in the area of software technologies [28].

Meanwhile, the unprecedented challenges and requirements arisen from BDA also drive revolutionary directions and opportunities of the software engineering discipline. It has been identified that dealing with the various Vs (such as volume, variety, velocity, veracity, etc.) of big data demands both novel functional features (e.g., new analytics algorithms and tools) and non-functional improvements (e.g., continuous delivery and quality assurance) of software systems in the BDA domain [16]. Thus, the association and interaction between software engineering and BDA-oriented data science will continue to foster software innovations.

1.3 Software-driven versus hardware-driven big data analytics

"Software-driven BDA" does not deny the value of hardware. There is no doubt that both the hardware infrastructure and the software stack fundamentally impact data analytics [16]. However, we argue that it is time to further strengthen and expand the role of software in BDA, mainly for three reasons.

Firstly, software as a driving force behind BDA has received less attention than it deserves. It is evident that a disproportionately larger amount of effort is being invested in the hardware infrastructure development over the software stack development in the BDA domain [21]. The imbalance between efforts on hardware and on software has been estimated to be as high as 80:20, while such a bias is clearly irrational, for their both fundamental impacts on analytical jobs. Worse still, such a bias might indicate

the existence of software crisis brewing in the big data ecosystem, not to mention that gigantic hardware resources could unexpectedly cause gigantic software problems [9]. Therefore, software deserves more attention even if it is equally as important as hardware in BDA implementations.

Secondly, hardware-driven BDA tends to become unsustainable. Currently many approaches (e.g., deep learning) in data science are computational resource hungry [29]. There is an increasing trend in employing more and more hardware resources (e.g., hundreds of GPU cards) to deal with big data problems. Unfortunately, those hardware-intensive solutions would be difficult to replicate, and could even lead to the Matthew effect or the monopoly of research and development in the community of data analytics [20], because most practitioners and academic researchers have little access to the industrial-sized clusters with thousands of computational nodes [33]. This has been noticed even by big BDA players. Although it is not a problem for them to afford heavy implementations, they have started advocating lightweight solutions based on software/algorithm breakthroughs, for instance Microsoft's LightLDA,[1] LightGBM,[2] and Google's MorphNet.[3] In addition, as stated by the Amdahl's Law [29], it is impossible to keep scaling hardware to address more and more sophisticated BDA problems. Even if the problems are 100% parallelizable or distributable, there will still appear software bottlenecks, as the infrastructural distribution inevitably increases the complexity and difficulty in both programming and deployment.

Thirdly, hardware is being softwarized. For various purposes ranging from reducing infrastructural cost to obtaining deployment agility and automation, there has arisen a disruptive trend in making the entire computing environment programmable and software-defined [18]. For example, software-defined network decouples the data transmission control from networking devices (e.g., switches and routers), software-defined storage separates the data store management from storage systems, and both of them leverage heterogeneous hardware to facilitate support of workload demands via open-interface programming [19]. The prospect is that the distinction between software and hardware will eventually vanish [10], as exemplified by "software as a medical device" [12]. Therefore, software-driven BDA is also the evolution direction towards softwarized infrastructure for deploying BDA applications.

2 Article overviews of this special issue

This special issue intends to explore the use cases, aspects/features, challenges, opportunities, and future directions associated with the practice and research in software-driven BDA. Here we provide an overview of the seven selected articles that represent a wide range of topics in this area.

(1) When hardware-intensive solutions are impractical, software strategies can help address the lack or bottleneck of hardware resources in BDA.

[1] https://www.microsoft.com/en-us/research/project/lightlda/.

[2] https://www.microsoft.com/en-us/research/project/lightgbm/.

[3] https://ai.googleblog.com/2019/04/morphnet-towards-faster-and-smaller.html.

Given the challenges in efficient statistical analysis of unlimited streaming big data events with limited storage, in the article entitled "Optimizing the confidence bound of count-min sketches to estimate the streaming big data query results more precisely" [13], the authors paid attention to parameter tuning of the probabilistic data structure count-min sketches. By employing an improved error measure based on binomial distribution and central limit theorem, there comes a tighter confidence bound that can make count-min sketches cost less time and storage, as well as improving their efficiency and accuracy.

(2) Emerging big data problems require inventions of innovative analytical methods and software solutions.

For example, given the large and open sets of remote sensing data generated by tons of satellite sensors nowadays, traditional analytical methods are no longer suitable for time-serial remote sensing data analysis that typically requires handling multidimensional spatio-temporal data models. In addition, it is tedious for practitioners and researchers to obtain ready-to-analyze data for Earth science models from raw observation data. In the article entitled "Spatial-feature Data Cube for spatiotemporal remote sensing data processing and analysis" [31], the authors developed a spatial-featured data cube tool for efficient time-serial remote sensing data processing and analysis. To obey the Amdahl's Law, the authors further employed a distributed execution engine for efficient implementation of large-scale tasks in parallel.

(3) The uncertainty and optimization problems in BDA generally rely on software solutions.

Within the edge cloud computing scenario, it is particularly challenging to allocate optimal cloud resources for real time analysis of big data streams from edge devices, if the data characteristics are unknown in advance. In the article entitled "Cloud resource management using 3Vs of Internet of Big data streams" [17], the authors proposed a novel method that could predict the data characteristics of streaming data in terms of volume, velocity and variety (3Vs), and then using Self-Organizing Maps (SOM) to arrange dynamic clusters of cloud resources. Note that, although cloud computing seems closely related to the hardware concepts (e.g., data center or computer farm), the virtualization behind cloud is essentially a software technology that creates an abstraction layer over the computing hardware layer.

(4) BDA has become a cornerstone to support many modern applications, and in other words, there must have been irreplaceable functional modules of BDA in those application systems.

For example, in the article entitled "Long-term real time object tracking based on multi-scale local correlation filtering and global re-detection" [34], the authors applied BDA techniques to visual object tracking that is one of the central research topics in the field of computer vision. In particular, the big data problem in this topic is due to the variation of the target and the surrounding environment. Correspondingly, a novel tracking algorithm based on local correlation filtering and global keypoint matching is proposed to solve problems occurred during long-term tracking such as occlusion, target-losing, etc.

(5) In a broad sense, the scope of software-driven BDA covers not only software engineering for/in BDA, but also BDA for/in software engineering.

Usability, as an essential software quality factor, is the degree to which a software product is employed by particular groups to achieve the goal of efficiency, effectiveness, satisfaction and many other features. In the article entitled "Software usability feature selection and evaluation using Modified Moth-Flame Optimization" [14], the authors developed a nature-inspired optimized algorithm called Modified Moth-Flame Optimization (MMFO) for usability feature selection. The MMFO algorithm reduces the number of features and retains a subset of relevant attributes without degrading the performances of the system.

(6) In addition to functionality and performance, security should also be one of the major concerns in software-driven BDA.

Data security and patient privacy are particularly crucial in the healthcare ecosystem. Since the healthcare industry has started adopting cloud to store personal health record (PHR), there is a need to ensure the ability of efficient search on encrypted data (stored in the cloud). In the article entitled "Secure search for encrypted personal health records from big data NoSQL databases in cloud" [5], the authors proposed a secure searchable encryption scheme, in order to search encrypted PHR from a NoSQL database in semi-trusted cloud servers. The proposed scheme supports almost all query operations available in plaintext database environments, especially the range query through multi-dimensional and multi-keyword searches.

(7) Besides the adjustable software behaviors and workloads at runtime, software product design, development and deployment all have influences on energy consumption [3]. Therefore, energy efficiency also deserves more attention in software-driven BDA.

The existing energy efficient scheduling methods of virtual machines (VMs) in the cloud cannot work well if the physical machines (PMs) are heterogeneous. In the article entitled "Implementation of an energy saving cloud infrastructure with virtual machine power usage monitoring and live migration on OpenStack" [32], the authors proposed a data-driven solution to an energy-efficient implementation of a cloud infrastructure. By monitoring the real-time status of virtual machines, this cloud implementation can automatically balance the virtual machines on every physical machine through live migration, as well as balancing the power consumption of every physical machine. Note that, although this article mainly focuses on the hardware-side energy consumption, we also suggest paying attention to the software energy efficiency in software-driven BDA.

3 Conclusion

Informed by the selected articles as well as our introduction, we conclude that software-driven BDA is a practical and booming domain that includes a broad range of research opportunities. We hope readers find our selection of articles interesting, and we expect this special issue to inspire both researchers and practitioners from multiple disciplines,

empiricists and theorists from relevant communities, to discuss the rigor, relevance, experience and challenges of this emerging domain. We also hope this special issue can attract more efforts to further develop a common research agenda for increasing the quality of current work and fostering collaborations between the software engineering community and the data science community.

Acknowledgements We appreciate all the authors who submitted their papers to our special issue, and we would also like to sincerely thank all of the reviewers who helped evaluate the submitted articles.

References

1. Bosch J (2017) Speed, data, and ecosystems: excelling in a software-driven world. Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series. CRC Press, Danvers, MA
2. Branstetter LG, Drev M, Kwon N (2018) Get with the program: software-driven innovation in traditional manufacturing. Manage Sci 65(2):541–558
3. Calero C, Piattini M (2015) Green in software engineering. Springer, Cham
4. Chasty C (2013) Forget the smart city... start with the smart workplace. https://www.wired.com/insights/2013/11/forget-the-smart-city-start-with-the-smart-workplace/. Accessed 24 May 2020
5. Chen L, Zhang N, Sun HM, Chang CC, Yu S, Choo KKR (2020) Secure search for encrypted personal health records from big data NoSQL databases in cloud. Computing 102(6): https://doi.org/10.1007/s00607-019-00762-z
6. Chen M, Mao S, Liu Y (2014) Big data: a survey. Mob Netw Appl 19(2):171–209
7. Cook DJ, Das SK (2007) How smart are our environments? An updated look at the state of the art. Pervasive Mob Comput 3(2):53–73
8. Dickinson R (2015) Using software for sustainability. https://en.muddyboots.com/news/view/using-software-for-sustainability. Accessed 24 May 2020
9. Ebert C (2018) 50 years of software engineering: progress and perils. IEEE Softw 35(5):94–101
10. Ebert C, Counsell S (2017) Toward software technology 2050. IEEE Softw 34(4):82–88
11. Fiore S, D'Anca A, Elia D, Palazzo C, Williams D, Foster I, Aloisio G (2014) Ophidia: a full software stack for scientific data analytics. In: Proceedings of the 12th international conference on high performance computing & simulation (HPCS 2014), IEEE Press, Bologna, Italy, pp 343–350
12. Gordon WJ, Stern AD (2019) Challenges and opportunities in software-driven medical devices. Nat Biomed Eng 3:493–497
13. Guo R, Xue E, Zhang F, Zhao G, Qu G (2020) Optimizing the confidence bound of count-min sketches to estimate the streaming big data query results more precisely. Computing 102(6). https://doi.org/10.1007/s00607-018-00695-z
14. Gupta D, Ahlawat AK, Sharma A, Rodrigues JJPC (2020) Feature selection and evaluation for software usability model using modified moth-flame optimization. 102(6). https://doi.org/10.1007/s00607-020-00809-6
15. Jacobs A (2009) The pathologies of big data. Commun ACM 52(8):36–44
16. Kambatla K, Kollias G, Kumar V, Grama A (2014) Trends in big data analytics. J Parallel Distrib Comput 74(7):2561–2573
17. Kaur N, Sood SK, Verma P (2020) Cloud resource management using 3Vs of Internet of Big data streams. Computing 102(6). https://doi.org/10.1007/s00607-019-00732-5
18. Li CS, Brech BL, Crowder S, Dias DM, Franke H, Hogstrom M, Lindquist D, Pacifici G, Pappe S, Rajaraman B, Rao J, Ratnaparkhi RP, Smith RA, Williams MD (2014) Software defined environments: an introduction. IBM J Res Dev 58(2/3):1–11
19. Li Z, Seco D, Rodríguez AES (2019) Microservice-oriented platform for internet of big data analytics: a proof of concept. Sensors 19(5), article no. 1134
20. Liu T (2019) Liu Tieyan talks about machine learning: there are too many easy followers, we need to reflect. https://www.msra.cn/zh-cn/news/features/tie-yan-liu-machine-learning. Accessed 24 May 2020

21. Madhavji NH, Miranskyy A, Kontogiannis K (2015) Big picture of big data software engineering: with example research challenges. In: Proceedings of the 1st international workshop on big data software engineering (BIGDSE 2015), IEEE Press, Florence, Italy, pp 11–14
22. Al-Jaroodi J, Hollein B, Mohamed N (2017) Applying software engineering processes for big data analytics applications development. In: Proceedings of the IEEE 7th annual computing and communication workshop and conference (CCWC 2017), IEEE Press, Las Vegas, NV, USA, pp 1–7
23. Penzenstadler B, Duboc L, Venters CC, Betz S, Seyff N, Wnuk K, Chitchyan R, Easterbrook SM, Becker C (2018) Software engineering for sustainability: find the leverage points!. IEEE Softw 35(4):22–33
24. Rabhi F, Bandara M, Namvar A, Demirors O (2018) Big data analytics has little to do with analytics. In: Beheshti A, Hashmi M, Dong H, Zhang WE (eds) ASSRI 2015, ASSRI 2017: service research and innovation. Lecture Notes in Business Information Processing, vol 234. Springer, Cham, pp 3–17
25. Roberts RD (2016) Why software really will eat the world—and whether we should worry. Indep Rev 20(3):365–368
26. Sachs R (2016) The mind as computer metaphor: Benson and the mistaken application of mental steps to software (part 3). https://www.bilskiblog.com/2016/04/the-mind-as-computer-metaphor-benson-and-the-mistaken-application-of-mental-steps-to-software-part-3/. Accessed 24 May 2020
27. Salit ML, Parsons ML (1985) Software-driven instrumentation: the new wave. Anal Chem 57(6):715A–729A
28. Spinellis D (2017) Future trends and research priorities in the area of software technologies. A report prepared for EU DG Communications Networks, Content and Technology PO 30-CE-0751856/00-91, European Commission, available online: https://ec.europa.eu/digital-single-market/en/news/future-trends-and-research-priorities-area-software-technologies. Accessed 24 May 2020
29. Stewart M (2019) The future of computation for machine learning and data science. https://towardsdatascience.com/the-future-of-computation-for-machine-learning-and-data-science-fad7062bc27d. Accessed 24 May 2020
30. Wang X (2019) Why the rise of software startup research: an insider's view. In: Hyrynsalmi S, Suoranta M, Nguyen-Duc A, Tyrväinen P, Abrahamsson P (eds) ICSOB 2019: software business. Lecture Notes in Business Information Processing, vol 370. Springer, Cham, pp 11–18
31. Xu D, Ma Y, Yan J, Liu P, Chen L (2020) Spatial-feature data cube for spatiotemporal remote sensing data processing and analysis. Computing 102(6). https://doi.org/10.1007/s00607-018-0681-y
32. Yang CT, Wan TY (2020) Implementation of an energy saving cloud infrastructure with virtual machine power usage monitoring and live migration on OpenStack. Computing 102(6). https://doi.org/10.1007/s00607-020-00808-7
33. Yuan J, Gao F, Ho Q, Dai W, Wei J, Zheng X, Xing EP, Liu TY, Ma WY (2015) LightLDA: Big topic models on modest computer clusters. In: Proceedings of the 24th international conference on world wide web (WWW 2015), International World Wide Web Conferences Steering Committee, Florence, Italy, pp 1351–1361
34. Zhao Q, Zhang B, Feng W, Du Z, Zhang H, Sun D (2020) Long-term real time object tracking based on multi-scale local correlation filtering and global re-detection. Computing 102(6). https://doi.org/10.1007/s00607-020-00807-8#citeas
35. Zhu X, Song B, Ni Y, Ren Y, Li R (2016) Software defined anything-from software-defined hardware to software defined anything. In: Business trends in the digital era. Springer, Cham, pp 83–103

Affiliations

Rajiv Ranjan[1] · Zheng Li[2] · Massimo Villari[3] · Yan Liu[4] · Dimitrios Georgeakopoulos[5]

✉ Rajiv Ranjan
raj.ranjan@ncl.ac.uk

✉ Zheng Li
imlizheng@gmail.com

Massimo Villari
mvillari@unime.it

Yan Liu
yan.liu@concordia.ca

Dimitrios Georgeakopoulos
dgeorgakopoulos@swinburne.edu.au

1 Newcastle University, Newcastle upon Tyne, UK

2 University of Concepción, Concepción, Chile

3 University of Messina, Messina, Italy

4 Concordia University, Montreal, Canada

5 Swinburne University of Technology, Melbourne, Australia