In Knowledge Management in Development of Data-intensive Software Systems, CRC Press/Taylor & Francis, Dec 2020

Foreword

John Grundy
Monash University
Melbourne, Australia
john.grundy@monash.edu

Introduction

Big data systems are becoming ubiquitous. The world produces a huge amount of data and software systems to make effective use of this data and required more than ever. However deciding what a data-intensive system should do, architecting and designing it, building and deploying it, and evolving it over time are all still very challenging activities [13].

Part of the cause of this challenge is the diverse knowledge needed to achieve an effective data-intensive system, the diverse team usually needed to develop and deploy the solution, and the ever-changing system and data landscape.

In this foreword I briefly characterise these challenges for data intensive systems that the chapters of this book address in various ways.

Big Data

The concept of 'big data' has become common place not only in software development but society [1]. We produce ever growing amounts of data that range from transport systems, health systems, energy infrastructure, social media, gaming, education, government services, leisure and tourism, scientific research, industry 4.0, and smart buildings and homes we live and work in [1,2,20]. Increasingly individuals, teams, organisations and governments want and indeed need to leverage this data to improve performance, security, outcomes, lifestyle, and wellbeing.

Several "V"s of big data – volume, variety, velocity, veracity, validity, volatility, value – are critical to support in any data-intensive system [7,16]. Volume typically refers to the size and complexity of data, which has grown almost exponentially in recent years. This includes social media data including video, text, images; health data including more precise genomics, MRI, etc capture; transport and energy data, including from smart grids and smart transport systems; sensor data from a wide range of Internet of Things-based systems increasingly collected from factories, buildings, hospitals and houses; and government data of many sorts. Velocity refers to increasing amounts, frequency of data from such systems. For example, modern vehicles have hundreds of sensors. Smart buildings thousands. Even individuals generate increasing data from wearables, smart homes, social media and work practices. Data comes in increasingly diverse varieties, in part due to the increasingly diverse systems generating the data. Combining and using this wide variety of data representations is an increasingly important yet challenging problem. Veracity concerns data quality, accuracy, reliability and the robustness of systems capturing, processing and storing data. Lack of trust in news media, sensor data, fake news, social media content, industrial control system data, security and privacy issues, and ethical use of data are all major challenges. Validity is critical to data chosen to solve the problem at hand. There is almost too much data in the world and care is needed to determine which combinations of datasets will actually address the data-intensive software system's user needs. Data is volatile in that while some data is stable e.g. geographic locale, topology, population

characteristics, physical structures etc, some is highly dynamic and changeable e.g. traffic and people movement, power consumption, individual health measurements, new social media data, etc. Finally, data analysis needs to add value – how can data be combined, new information and knowledge mined, new insights be derived, to enable the data-intensive system to aid its users.

Data-intensive Software Systems

Data-intensive software systems are characterised by their high dependence on diverse data that is critical to the system's functionality but also many non-functional issues [6,13,22]. Data has to be sourced from a wide range of sensors, devices, other systems. This might include range of IoT sensors; health systems; energy, transport, utility, grid etc systems; building management systems; variety of government systems; Geographic Information Systems; industrial control systems; personal sensors; and social media systems [10]. Some data is simple and low volume, but much is complex and high volume and frequency. This means the data-intensive system may need to do a lot of processing of the data to turn it into a form that is useful.

Data from diverse sources needs to be integrated and "harmonised", to link up similar/same data items and produce a unified new set of data for further processing and usage [3]. This activity can include data wrangling, format changes, merging, splitting, joining data, and – often complex and imprecise – harmonising similar concepts, terms, formats and ontologies distributed across diverse source data sets.

Integrated data may need to be stored, but some large, complex or restricted access data might need to be retrieved as needed from source systems [5]. Network constraints may impact data-intensive system performance. A variety of data processing are typically needed – machine learning, pattern recognition, information retrieval and other techniques used to find information and extract knowledge from integrated datasets [25].

Decision support is a critical aspect of most data-intensive systems. Decisions may vary from large scale traffic analysis and control, smart building management systems, smart hospitals, industry 4.0 control rooms, and government policy making, to individual and team decision making, including AI-supported project management, smart homes, travel planning and health and welling decision making [9]. Many of these need to be supported by complex data visualisation systems, presented analysed datasets in forms end users can interpret and make use of.

Software Engineering for Data-intensive Systems

A range of challenging software engineering issues present in building and maintaining such data-intensive systems [6,15]. What process should be used is an interesting question – many projects have adopted Agile techniques and at first glance this would seem a good fit where the range of data sources, data processing and use of data may vary over time. But many data-intensive systems are safety and security critical, and having many iterations, refactorings, and sprint-based delivery might not be the best fit approach.

Identifying requirements for data-intensive systems is a challenge as both end users and data sources are likely to be volatile over the life of the project development and evolution. This means requirements may be quiet emergent – new data sources formats, greater volume,

frequency and variable quality (improve or even reduce) may all significantly impact the system under development. Non-functional requirements can be very challenging. Due to the disparate data size, frequency, variety and quality of data this puts a lot of very high demands on systems. This includes performance and response time constraints; reliability and robustness where connected systems may be unpredictable in availability and their own performance; rapidly changing data volume and quality; updated connectors to source data; updated platforms hosting data-intensive systems and processing algorithms; new users with new visualisation and data processing requirements; and evolving security and privacy requirements.

A data-intensive system often has extensive architecting and design challenges [22,23]. In order to interface to diverse data sources it needs to realise a range of technology connectors. In order to retrieve, filter/wrangle, transform, integrate, harmonize and store all/part of source system data, a range of data processing and management technologies may be required. In order to analyse collected data, a range of machine learning, information retrieval, indexing, natural language processing, image processing, and other advanced techniques, algorithms, platforms, and solutions need to be used. Visualisation solutions may require extensive UI design and implementation effort, and depending on the technology desired by users, may also require significant platform resources e.g. for VR-based support [4].

Knowledge Management

Data-intensive systems are usually built not by software engineers alone but teams also include domain experts, data scientists, organisational managers, and cloud/compute platform engineers [19]. Such a multi-disciplinary team puts a lot of demands on not only software engineering process and project management, but also knowledge management.

Current approaches to capturing, evaluating and using requirements for data-intensive systems are not well suited to such multi-disciplinary teams. Approaches used are often focused on one or two stakeholder groups and do not suit or fit the needs of others. There are no agreed standards to describe data, data processing, and data visualisation [19]. The architecture and implementation of data-intensive systems are necessarily often very complex and multi-stakeholder input is even needed to engineer the system e.g. AI experts, software engineers, cloud and grid computing experts, IoT experts, and database experts. Best practices for creating and sharing such diverse knowledge across such a team are still being developed for data-intensive software system engineering [21].

As noted above, due to the several Vs in big data domain, data-intensive systems inherently live in a changing environment. New data is made available. Data quality, volatility and veracity change. Validity and value of data used may change as stakeholder needs change. Most data-intensive systems exhibit various degrees of emergent requirements, where these new/changed data sources and new/changed stakeholder needs severely impact on the system in many ways. Handling these emergent requirements is still extremely challenging [8].

Other concerns

Many other issues present when engineering next-generation, data-intensive systems. Security and privacy are increasing challenges. Users expect data to be collected and used for specific purposes, but the interconnectedness of systems, and ability to transfer sensitive data from one system to another and lose data provenance exposes many privacy concerns [11,24]. Many

data-intensive systems are security-critical and safety-critical, in that they deal with utility, transport, health, manufacturing, and other high value, high criticality system domains. Due to the rapid evolution of data intensive systems brought on by the changing nature of the underlying big data domain, many technical debt challenges present [12]. Choosing particular approaches to data sourcing, wrangling, storage, processing and visualisation may seem appropriate at one time, but then incur a variety of serious technical debt implications down the track. Ethics and wider human values issues relating to data-intensive systems represent important new areas of research and practice. Socio-technical issues present an interesting challenge in this domain. As well as the multi-disciplinary team, data-intensive systems are often used by a wide variety of stake-holders e.g. citizens, patients. They have a wide range of diverse human factors impacting on their likely ability to use and take up the system that needs approaches to adequately incorporate into development and evolution [17]. Finally, due again to the nature of the big data domain and its volatility, data-intensive systems are almost never "finished", with new data sources, changes to data availability, quality and volume being inherent in the domain. This makes evolving data-intensive systems even more challenging than conventional software systems [8,14].

This book provides diverse chapters addressing many of the outstanding issues in the domain of knowledge management for data-intensive software system engineering. I do hope that you find them helpful in your understanding and development of next-generation software-intensive systems!

References

1. Al Nuaimi, E., Al Neyadi, H., Mohamed, N., & Al-Jaroodi, J. (2015). Applications of big data to smart cities. *Journal of Internet Services and Applications*, *6*(1), 25.
2. Al-Ali, A. R., Zualkernan, I. A., Rashid, M., Gupta, R., & AliKarar, M. (2017). A smart home energy management system using IoT and big data analytics approach. IEEE Transactions on Consumer Electronics, 63(4), 426-434.
3. Avazpour, I., Grundy, J., & Zhu, L. (2019). Engineering complex data integration, harmonization and visualization systems. *Journal of Industrial Information Integration*, *16*, 100103.
4. Benzaken, V., Fekete, J. D., Hémery, P. L., Khemiri, W., & Manolescu, I. (2011). EdiFlow: data-intensive interactive workflows for visual analytics. In *2011 IEEE 27th International Conference on Data Engineering* (pp. 780-791). IEEE.
5. Cai, H., Xu, B., Jiang, L., & Vasilakos, A. V. (2016). IoT-based big data storage systems in cloud computing: perspectives and challenges. *IEEE Internet of Things Journal*, *4*(1), 75-87.
6. Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information sciences*, *275*, 314-347.
7. Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile networks and applications*, *19*(2), 171-209.
8. Cleve, A., Mens, T., & Hainaut, J. L. (2010). Data-intensive system evolution. Computer, (8), 110-112. IEEE.
9. Demirkan, H., & Delen, D. (2013). Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decision Support Systems*, *55*(1), 412-421.
10. Dong, X. L., & Srivastava, D. (2013). Big data integration. In *2013 IEEE 29th international conference on data engineering (ICDE)* (pp. 1245-1248). IEEE.
11. Fernandez, E. B. (2011). Security in data intensive computing systems. In *Handbook of Data Intensive Computing* (pp. 447-466). Springer, New York, NY.
12. Foidl, H., Felderer, M., & Biffl, S. (2019). Technical Debt in Data-Intensive Software Systems. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 338-341). IEEE.
13. Furht, B., & Escalante, A. (Eds.). (2011). *Handbook of data intensive computing*. Springer Science & Business Media.
14. Goeminne, M., Decan, A., & Mens, T. (2014). Co-evolving code-related and database-related changes in a data-intensive software system. In *2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)* (pp. 353-357). IEEE.
15. Gorton, I., Greenfield, P., Szalay, A., & Williams, R. (2008). Data-intensive computing in the 21st century. *Computer*, *41*(4), 30-32.

16. Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big data: Issues and challenges moving forward. In *2013 46th Hawaii International Conference on System Sciences* (pp. 995-1004). IEEE.
17. Kato, J., Igarashi, T., & Goto, M. (2016). Programming with examples to develop data-intensive user interfaces. *Computer*, *49*(7), 34-42.
18. Khalajzadeh, H., Abdelrazek, M., Grundy, J., Hosking, J. G., & He, Q. (2019). Survey and Analysis of Current End-user Data Analytics Tool Support. *IEEE Transactions on Big Data*.
19. Khalajzadeh, H., Simmons, A., Abdelrazek, M., Grundy, J., Hosking, J., & He, Q. (2020). An End-to-End Model-based Approach to Support Big Data Analytics Development. *Journal of Computer Languages*, 100964.
20. Kim, G. H., Trimi, S., & Chung, J. H. (2014). Big-data applications in the government sector. *Communications of the ACM*, *57*(3), 78-85.
21. Kleppmann, M. (2017). *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. " O'Reilly Media, Inc.".
22. Mattmann, C. A., Crichton, D. J., Medvidovic, N., & Hughes, S. (2006). A software architecture-based framework for highly distributed and data intensive scientific applications. In *Proceedings of the 28th international conference on Software engineering* (pp. 721-730).
23. Mattmann, C. A., Crichton, D. J., Hart, A. F., Goodale, C., Hughes, J. S., Kelly, S., and Medvidovic, N. (2011). Architecting data-intensive software systems. In *Handbook of data intensive computing* (pp. 25-57). Springer, New York, NY.
24. Smith, M., Szongott, C., Henne, B., & Von Voigt, G. (2012) Big data privacy issues in public social media. In *2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)* (pp. 1-6). IEEE.
25. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... & Ghodsi, A. (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM*, *59*(11), 56-65.