Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/75/20240518

Challenges and strategies of software engineering in big data environment

Bruce Cheng

School of Software Engineering, Qingdao University of Science and Technology, Shandong, China

18265461328@163.com

Abstract. Software engineering in a big data environment involves developing and maintaining large-scale, scalable, data-intensive applications in a rapidly evolving technology domain. As data volumes grow exponentially, so do the requirements for software systems to process and analyze this data. These systems need to be able to efficiently store, retrieve, process and analyze data collected from sources such as social media, IoT devices, enterprises and scientific research. Software engineers must address challenges including data heterogeneity and quality, scalability of data storage and computation, and real-time processing of high-speed data streams. With the evolution of technology, new architecture and database technologies have emerged, such as NoSQL and distributed computing frameworks. These technologies have challenged traditional methods of software engineering practice and also brought new design and implementation strategies.

Keywords: Software engineering, big data, challenge, strategy.

1. Introduction

1.1. Overview of software engineering in big data environment

In the field of software engineering, the introduction of the concept of big data is an innovation to traditional practices. This involves collecting, managing, analyzing and interpreting large, rapidly generated data sets. Big data is not only huge in volume, but also complex and varied, containing multiple types and formats of data, challenging the ability to store, search, share, visualize and analyze. Basic concepts of software engineering, such as modularization, abstraction, reuse, and maintenance, need to be rethought in the context of big data[1]. Systems that handle large-scale data sets need to be able to quickly adapt to new technologies and changing business needs while maintaining efficiency and reliability.

2. Introduction to basic concepts

2.1. Big Data

Big data refers to a data set that is so large and complex that it is difficult for traditional data processing applications to effectively process it. Big data not only refers to the amount of data, but also involves various characteristics of data, which are usually summarized as the "five Vs":

DOI: 10.54254/2755-2721/75/20240518

Volume: One of the core characteristics of big data is its huge volume. The amount of data can range from several TB (Terabytes) to several PB (Petabytes) or even more.

Velocity: The speed of data flow or the speed at which data is generated. In a big data environment, data is generated and collected at extremely fast speeds. For example, the amount of data generated by social media sites every day is a good example.

Variety: Data comes from multiple sources and exists in multiple formats. These formats can be structured (such as databases), semi-structured (such as XML files), or unstructured (such as text, video, audio).

Veracity: the quality and reliability of data. Big data involves information from a variety of sources, which may be incomplete, inaccurate, or biased.

Value: Extracting valuable information from big data is a major challenge. The large amount of data itself is not the point; what is important is the insights and information gained from analyzing this data[1].

Big data has a wide range of applications, including but not limited to business intelligence, predictive analytics, data science, and machine learning. Through efficient big data analytics, organizations can discover patterns, trends, and correlations hidden in large amounts of data, which are critical for decision-making, predicting trends, optimizing operations, and improving user experience. The application of big data technology has penetrated into many fields such as healthcare, financial services, retail, manufacturing, government and education.

2.2. Data Lake

A data lake is a system or repository that stores large-scale data. It can store large amounts of raw data until it is needed. Unlike traditional data warehouses, data lakes can store unstructured, semi-structured and structured data, providing highly flexible and scalable storage solutions. Here are some key features of a data lake:

Diverse data formats: Data lakes can store data in various formats, including text files, images, videos, audios, log files, XML, JSON, CSV, etc.

Storage of raw data: Unlike traditional data warehouses, data lakes store raw data, whether structured or unstructured. This means that the data can be stored without pre-processing.

Scalability and cost-effectiveness: Data lakes are often built on cheap hardware or cloud infrastructure, making them highly scalable and cost-effective.

Data governance and security: Data lakes require an effective data governance strategy to ensure data quality, security, and compliance[2].

The use of data lakes is becoming more and more common in different fields, especially in scenarios where large amounts of data need to be processed and analyzed, such as big data analysis, machine learning projects, data science research, etc. As technology develops, the concept of data lakes continues to evolve, gradually integrating more data management and analysis tools to provide more comprehensive data solutions.

2.3. Distributed computing

Distributed computing is a computing method that distributes data processing across multiple computers.

2.4. Data Mining

Data mining is the process of extracting (mining) useful information and knowledge from large data sets.

2.5. Machine Learning

Machine learning is the use of algorithms and statistical models to enable computer systems to make predictions or behavioral decisions based on data without the use of explicit instructions.

DOI: 10.54254/2755-2721/75/20240518

2.6. Cloud computing
Cloud computing is the provision of on-demand computing resources and data storage services through the Internet[3].

2.7. NoSQL databases
NoSQL is a mechanism for storing and retrieving data, and its model construction does not follow the traditional relational model[4].

2.8. Real-time processing
Real-time processing is the process of processing and analyzing data immediately after it is entered into the system.

2.9. Scalability
Scalability is the ability of a system to expand its processing capabilities as demand increases.

The above concepts are the cornerstones for understanding software engineering practices in big data environments. They will be expanded in this article and their specific applications in the design and implementation of large-scale data-intensive systems will be discussed in detail.

3. Challenges of software engineering in big data environment
In the big data environment, the core challenges faced by software engineering mainly include the following aspects:

3.1. Data management and processing
Processing of massive data sets from disparate sources, often heterogeneous and in various formats. This includes not only data collection and storage, but also complex data processing processes[5]. For example, the use of data lakes and NoSQL databases poses new challenges, and how to effectively organize and access data in these systems becomes key.

In terms of heterogeneous data integration, big data environments usually cover structured data, semi-structured data and unstructured data. Software engineers need to design systems that can integrate these different data sources to ensure that the data can be processed uniformly.

3.2. Performance and scalability
As data volumes grow, maintaining system performance and responsiveness becomes increasingly difficult. In this context, cloud computing and distributed computing have become key technologies that can provide the necessary computing resources and storage capabilities to support the growing demand for data processing[6].

In a big data environment, data processing must be efficient. This requires software engineers to consider data indexing, query optimization, storage structure and data processing algorithms when designing the system to improve performance.

3.3. Data security and privacy
When dealing with large amounts of sensitive data, protecting privacy and preventing data breaches is crucial. This requires complex security mechanisms and policies to ensure the safety of data during storage and transmission.

Data security and privacy protection are also another important aspect of data management. Various security measures such as authentication, authorization, encryption, and desensitization need to be implemented[7].

DOI: 10.54254/2755-2721/75/20240518

3.4. Real-time data processing

Another challenge in a big data environment is processing and analyzing data streams in real-time. This requires systems that can collect and process data instantly to support rapid decision-making and response.

Many business scenarios require real-time data analysis, which places strict requirements on the response practice of data processing. Stream processing platforms such as Apache Kafka and Apache Flink are used to support this requirement[8].

3.5. Technology selection and adaptability

With the rapid development of technology, choosing the right technology stack and tools is becoming increasingly important. Software engineers need to constantly evaluate new technologies and quickly adapt to these changes to fully exploit the potential of big data.

3.6. Cost Control

The cost of storing and processing large amounts of data can be very high. Therefore, measures need to be taken to optimize resource usage, such as reducing costs through data lifecycle management and timely data storage tiering.

The above challenges require software engineers to continuously learn and adapt to new technologies, and they also need to take these factors into consideration during the design and implementation stages to build efficient, secure, and scalable big data systems. These challenges incorporate expertise across disciplines, including data science, software engineering, systems architecture, and cybersecurity. As technology develops, these challenges continue to evolve, requiring practitioners to continually learn and adapt to new technologies and methods.

4. Strategies of software engineering in big data environment

In response to the above challenges, the following are some specific coping strategies:

4.1. Adopt efficient data management technology

For data management and processing, we can use data lakes to store large amounts of raw data, and utilize advanced data processing frameworks such as Hadoop or Spark to effectively process this data.

4.2. Achieve system scalability

Using cloud computing and distributed computing technology, we can design a system that can dynamically scale according to demand, thereby maintaining high performance as the amount of data grows[9].

4.3. Strengthen data security and privacy protection

Apply encryption technology, access control and data masking technology to protect data security and privacy.

4.4. Utilize real-time data processing technology

Use stream processing technology and real-time analysis tools, such as Apache Kafka and Apache Storm, to process and analyze real-time data[9].

4.5. Continuous technology evaluation and selection

Continue to pay attention to technology development and select a technology stack that suits specific needs, such as NoSQL databases to process unstructured data.

4.6. Adopt agile and iterative development methods

In the rapidly changing big data environment, agile development methods can help teams adapt to changes faster and continuously improve software products.

DOI: 10.54254/2755-2721/75/20240518

4.7. Promote cross-disciplinary collaboration

Big data projects often require cross-domain knowledge and skills. Software engineers should work closely with professionals such as data scientists and business analysts to solve problems together.

These strategies combine key technologies and methods in the big data environment, aiming to solve specific challenges and achieve efficient, secure, and scalable data analysis.

5. Application scenarios-taking Amazon as a case

In reality, some large retail companies are faced with how to manage and analyze their huge customer data. Companies typically use data lakes to store and organize various data formats. In response to the challenges of performance and scalability, engineers chose to deploy a cloud-based distributed computing system that can dynamically expand according to data processing needs. To ensure data security, the company implements necessary multiple layers of encryption and strict access control measures. Real-time data processing is achieved through the introduction of stream processing technology, allowing companies to analyze customer behavior in real time. Finally, ongoing technology assessment ensures the company can leverage the latest NoSQL databases and big data analytics tools to stay ahead of the curve[10]. Through these strategies, companies can successfully solve many challenges in data management and analysis, further improve business efficiency and enhance customer satisfaction.

Take Amazon as an example. The company has a huge set of customer data, and its method of managing and analyzing this data is a good example of the above application scenarios:

Amazon uses data lakes to store data in multiple formats, including customer transaction records, browsing history, product reviews, etc. They utilize services in AWS (Amazon Web Services) such as Amazon S3 as a storage solution for the data lake to organize this data.

For performance and scalability, Amazon deploys distributed computing services such as Amazon Redshift and Amazon EMR. These services allow Amazon to efficiently process and analyze large amounts of data by automatically scaling based on data processing needs.

For data security, Amazon implements end-to-end encryption during data transmission and storage, and strictly controls data access permissions to ensure that only authorized users can access sensitive data[11].

In terms of real-time data processing, Amazon uses stream processing technologies such as Amazon Kinesis, which allows them to collect, process and analyze customer data in real time in order to quickly respond to market changes and personalize customer experience.

Finally, Amazon regularly evaluates and adopts new technologies, such as NoSQL databases (such as Amazon DynamoDB) and big data processing tools (such as Apache Spark), to ensure that they remain at the forefront of data processing and analysis[12].

Through such a strategy, Amazon is able to effectively process and analyze its massive customer data sets to provide customers with a personalized shopping experience while optimizing inventory management and operational efficiency.

6. Future Outlook

In the future where big data technology continues to develop, we can foresee that software engineering will pay more attention to data-driven decision-making processes and intelligent system design. Further improvements in distributed computing and cloud technology will make processing large-scale data sets more efficient. Data security and privacy protection will be a major focus, and as regulations evolve, companies will need to adopt more advanced security measures. Real-time data analysis will become more important, and companies will rely more on instant insights to respond quickly to market changes. In addition, the integration of machine learning and artificial intelligence will make data analysis more precise and automated, bringing deeper insights and enhanced competitive advantages to enterprises[13].

At the same time, software engineering in the big data environment will further integrate concepts such as data lakes, cloud computing, and distributed computing. We can expect the application of

DOI: 10.54254/2755-2721/75/20240518

more advanced NoSQL databases and real-time data processing technology to make data analysis more efficient and accurate[14]. With the integration of machine learning and artificial intelligence, data-driven decision-making will become more automated and intelligent. Data security and privacy protection will also become key areas, and new technologies and regulations will continue to emerge to address these challenges. Overall, software engineering will focus more on flexibility, scalability, and security to adapt to increasingly complex and dynamic big data environments.

7. Conclusion

We comprehensively explore the main challenges of software engineering in big data environments, such as data management, performance scalability, data security, real-time processing, and technology selection. In response to these challenges, we propose strategies including leveraging distributed and cloud computing, enhancing data security, enabling real-time data processing, and continuous technology adaptation. We present the practical application of these strategies through the experience of a retail company. Looking to the future, as technology develops, more efficient and automated data processing methods are expected to emerge, while data security and privacy protection will become a core focus.

References

[1] Wu, J., Guo, S., Li, J., & Zeng, D. (2016). Big data meet green challenges: Greening big data. IEEE Systems Journal, 10(3), 873-887.

[2] Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. Journal of business research, 70, 263-286.

[3] Fernández, A., del Río, S., López, V., Bawakid, A., del Jesus, M. J., Benítez, J. M., & Herrera, F. (2014). Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(5), 380-409.

[4] Grüning, B. A., Lampa, S., Vaudel, M., & Blankenberg, D. (2019). Software engineering for scientific big data analysis. GigaScience, 8(5), giz054.

[5] Davoudian, A., & Liu, M. (2020). Big data systems: A software engineering perspective. ACM Computing Surveys (CSUR), 53(5), 1-39.

[6] Wu, J., Guo, S., Li, J., & Zeng, D. (2016). Big data meet green challenges: Big data toward green applications. IEEE Systems Journal, 10(3), 888-900.

[7] Hu, J., & Vasilakos, A. V. (2016). Energy big data analytics and security: challenges and opportunities. IEEE Transactions on Smart Grid, 7(5), 2423-2436.

[8] Rodríguez-Mazahua, L., Rodríguez-Enríquez, C. A., Sánchez-Cervantes, J. L., Cervantes, J., García-Alcaraz, J. L., & Alor-Hernández, G. (2016). A general perspective of Big Data: applications, tools, challenges and trends. The Journal of Supercomputing, 72, 3073-3113.

[9] Asch, M., Moore, T., Badia, R., Beck, M., Beckman, P., Bidot, T., ... & Zacharov, I. (2018). Big data and extreme-scale computing: Pathways to convergence-toward a shaping strategy for a future software and data ecosystem for scientific inquiry. The International Journal of High Performance Computing Applications, 32(4), 435-479.

[10] Hamad, F., Fakhuri, H., & Abdel Jabbar, S. (2022). Big data opportunities and challenges for analytics strategies in Jordanian Academic Libraries. New Review of Academic Librarianship, 28(1), 37-60.

[11] Gupta, A., Agarwal, D., Tan, D., Kulesza, J., Pathak, R., Stefani, S., & Srinivasan, V. (2015, May). Amazon redshift and the case for simpler data warehouses. In Proceedings of the 2015 ACM SIGMOD international conference on management of data (pp. 1917-1923).

[12] Hummel, O., Eichelberger, H., Giloj, A., Werle, D., & Schmid, K. (2018, August). A collection of software engineering challenges for big data system development. In 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 362-369). IEEE.

DOI: 10.54254/2755-2721/75/20240518

[13] Biesialska, K., Franch, X., & Muntés-Mulero, V. (2021). Big Data analytics in Agile software development: A systematic mapping study. Information and Software Technology, 132, 106448.
[14] Otero, C. E., & Peter, A. (2014). Research directions for engineering big data analytics software. IEEE Intelligent Systems, 30(1), 13-19.