UDC 004.652
DOI: 10.15587/2706-5448.2022.251281
Article type «Reports on Research Projects»

Sofiia Materynska,
Vadym Yaremenko,
Walery Rogoza

A THEORETICALLY PROPOSED ALGORITHM IN A DECISION TREE FORMAT FOR CHOOSING AN EFFICIENT STORAGE TYPE OF LARGE DATASETS

The object of research is methods and approaches to improve storage efficiency and optimize access to large amounts of data. The importance of this study consists in the wide dissemination of big data and the need for the right selection of technologies that will help improve the efficiency of big data processing systems. The complexity of the choice is caused by the large number of different data storages and databases that are available now, so the best decision requires a deep understanding of the advantages, disadvantages and features of each. And the difficulty lies in the lack of a universal algorithm for deciding on the optimal repository. Accordingly, based on the experiments, analysis of existing projects and research papers, a decision-making algorithm was proposed that determines the best way to store large datasets, depending on their characteristics and additional system requirements. This is necessary to simplify the design of the system in the early stages of big data processing projects. Thus, by highlighting the key differences, as well as the disadvantages and advantages of each type of storage and database, a list of key characteristics of the data and the future system, which should be considered when designing.

This algorithm is a theoretical proposal based on the studied research papers. Accordingly, using this algorithm at the design stage of the system, it would be possible to quickly and clearly determine the optimal type of storage of large datasets. The paper considers column-oriented, document-oriented, graph and key-value types of databases, as well as distributed file systems and cloud services.

Keywords: *large datasets, non-relational database, column-oriented database, document-oriented database, key-value database, graph database.*

Received date: 16.10.2021
Accepted date: 14.12.2021
Published date: 19.01.2022

How to cite
Materynska, S., Yaremenko, V., Rogoza, W. (2022). A theoretically proposed algorithm in a decision tree format for choosing an efficient storage type of large datasets. Technology Audit and Production Reserves, 1 (2 (63)), 6–9. doi: http://doi.org/10.15587/2706-5448.2022.251281

1. Introduction

Every year, there is a significant increase in the amount of data produced by large companies, enterprises, governments, and ordinary people. This data has different sources of origin and is related to different fields: healthcare, economics, marketing, business, and others. Such huge amounts of data are the result of expanding the use of social networks, electronic devices, and other information technologies. Since the opportunity to produce large amounts of data came, new perspectives in the processing and wise usage of this data appeared. Big data analytics bring great benefits and significant revenue to businesses every day through its various applications.

However, to benefit from the use of big data technologies, they need to be effectively applied. One of the main problems with application of big data technologies is data storage [1], as in addition to storing data in a certain form, it is also necessary to provide it with appropriate access to meet the needs of the next stages of use of this data. For example, after pre-processing and cleaning raw data intended for machine learning, artificial intelligence processing or analytics, it is necessary to place it in a specific data storage or database that will be most effective in a particular situation [2].

The object of research includes methods and approaches to improve storage efficiency and optimize access to large amounts of data. *The aim of this research* is a comparative analysis of big data storage methods and creation of a decision support system to determine the most appropriate method.

2. Research methodology

In this study, various publications related to methods of storing and accessing big data [3, 4] were analysed. As a result of the analysis, the most efficient and widespread storage systems for different requirements were identified. These storage systems include distributed file systems [5], cloud services, non-relational databases: key-values, column-oriented, document-oriented and graph. In common case, relational databases are not used in big data projects as they cannot provide high availability and horizontal scalability unlike NoSQL databases [6, 7].

Each of the considered approaches has its advantages, disadvantages, and features that determine for which type of data they are best suited. Key-value databases provide efficient storage of unstructured data, and quick access to write through a unique key, but they do not allow the use of SQL-like queries. Such databases do not work well with frequently updated data but are very scalable.

Column-oriented databases are well suited for further work with analytical systems, as they support SQL-queries. They are important for structured data because they support easy-to-modify schemas, and column-oriented databases support a high level of data compression associated with the storage approach. Column-oriented databases also support great scalability.

Document-oriented databases provide a hierarchical system based on the key-value principle and allow the storage of unstructured and semi-structured data, usually in JSON-like format. They support easy data uploading and updating and simple querying of the content of documents, as well as scalability and indexing.

Graph databases are well-suited for data that assumes that relations between records are present and important. They represent all the data as a graph with nodes, edges and their properties and support a graph traversal, but they are less scalable and require a lot of effort to maintain.

Distributed file systems can be used to efficiently store raw data that needs to be processed. They can store all the different formats. There are tools that offer queries to semi-structured or unstructured data so that they can be used to work with such data storage. Today, various cloud services offer a wide range of tools for big data storing and processing. The use of such services can greatly simplify the deployment and maintenance of the system. They also provide high scalability, ongoing support, and data security [7, 8]. Efficient strategies for storing large datasets were considered in [9], however, this work doesn't provide a general approach for database choose in production projects. At that time in [10] a security-oriented way is described for solving the similar problem, but this work can be also improved by proposing a general way for choosing an optimal database. As the result this work is about developing a general algorithm for decision making during the project architecture design stage.

3. Research results and discussion

In the process of developing a big data processing system, the question arises as to which repository is best suited for a particular case. In this study was developed a decision tree (Fig. 1–3) that allows to determine which method of data storage may be potentially most effective.

The tree nodes are certain statements that reflect the characteristics of the data or requirements for the storage system, the edges correspond to possible answers – yes or no. Leaves represent a type of storage or a specific database.

In the decision-making process, it is proposed to determine: the degree of data processing, variety of formats, the presence and importance of relations between records, data structure, the need to support queries, write and read intensity, scalability, and access speed.

While deciding, the first step is to define if data that will be stored is raw (Fig. 1). If it is true, and it arrives in different formats then it may be a good decision to use a distributed file system, but the better choice will be a cloud storage as it provides scalability and stable access without any failures or data loss. In case, data is already processed, or only one data format is collected, let's move to analysis of relations between records. If such relations are present, important, and will be considered in further data analysis then probably graph databases can fit well.

In case there are no relations between records, it should be defined if data is structured or not (Fig. 2). If it is and the system will require intensive reading together with significant query support, then it would be a good idea to use a column-oriented database (DB) as it meets the requirements and can be used even for complex analytics. When with the same conditions, query support is not that necessary, a document-oriented database can be chosen.

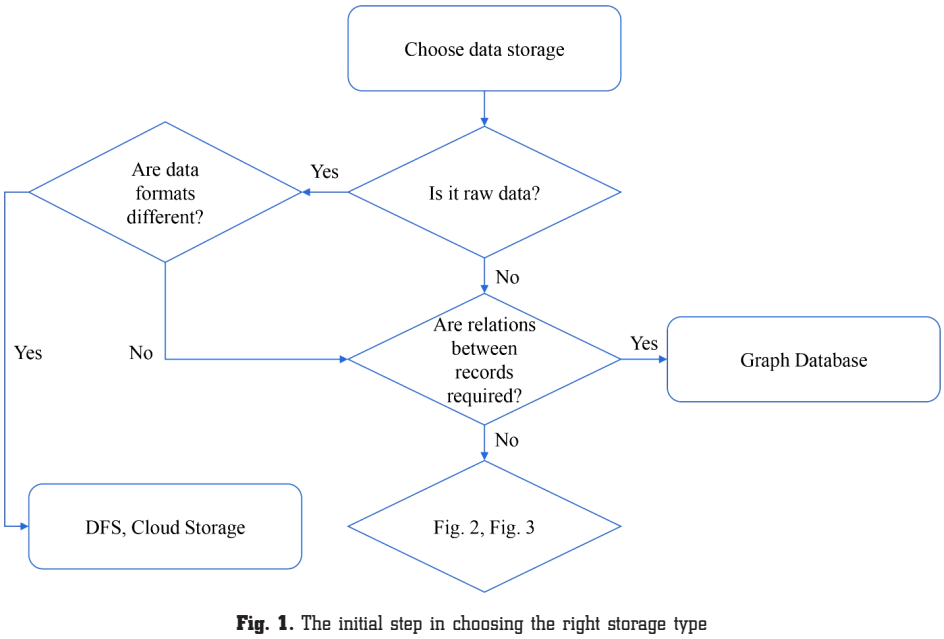

Fig. 1. The initial step in choosing the right storage type

ISSN 2664-9969

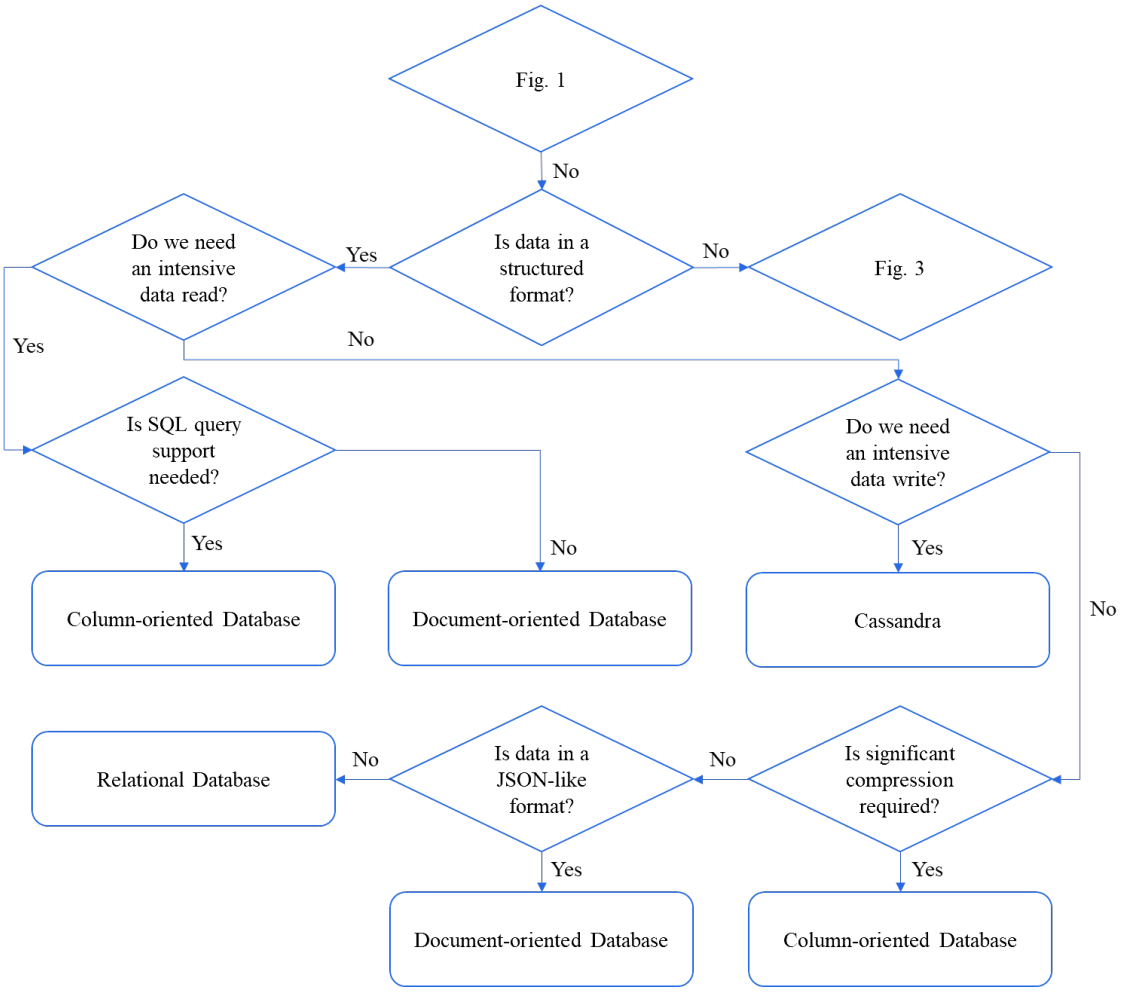

Fig. 2. Possible choose of an efficient data storage for structured data

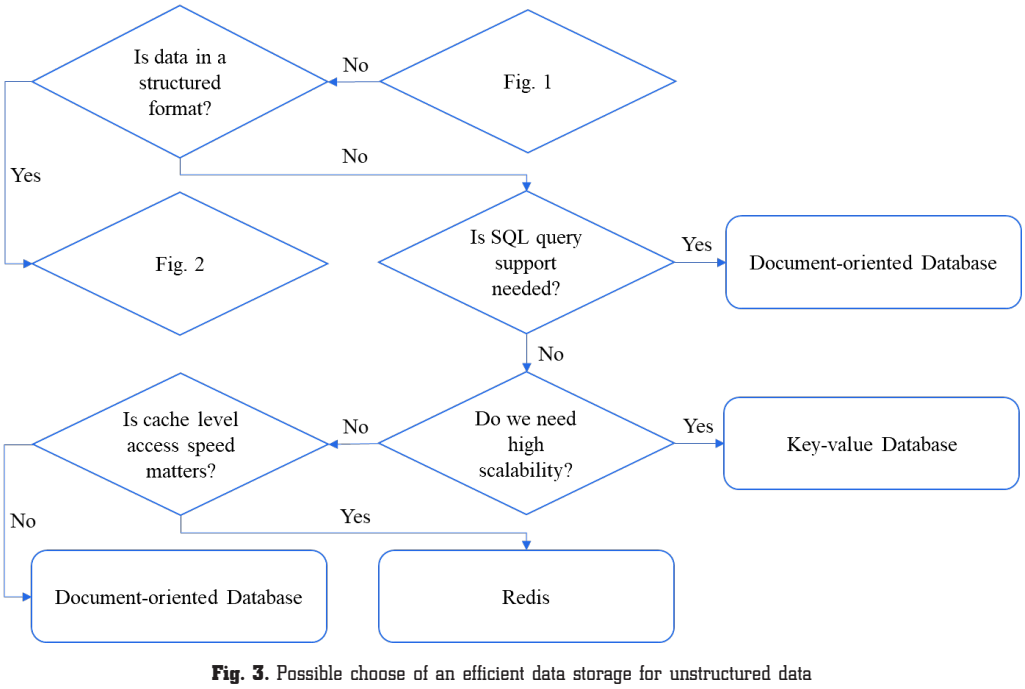

Fig. 3. Possible choose of an efficient data storage for unstructured data

If there are requirements on intensive write not intensive read, then Apache Cassandra may fit well as it combines advantages of column-oriented and key-value databases. Otherwise, it should be determined if significant data compression is required due to huge data volumes. If so, a column-oriented store may be used and in the

opposite case when data has JSON-like or XML-like format, then document-oriented DB like MongoDB may be a good choice. If none of these conditions is relevant, probably there is no need to use non-relational databases or storage, as data can be stored in some relational DB.

Starting now from the step that considers the definition of presence of data structure, let's move on to semi-structured and unstructured data (Fig. 3). If querying is not necessary but high scalability is, then pay attention to key-value databases as it does not support queries but works well with unstructured data, stored by unique key, can be easily scaled, and quickly accessed. If scalability is not important, but access time should be even smaller, try using Redis as it stores data in RAM. Otherwise, document-oriented DB can be used as it stores semi-structured and unstructured data and supports querying.

The obtained algorithm is a theoretical proposal based on the studied research papers. The results of this work can be used to create a program that will offer the best storage for the user based on its requirements. It is also possible to test many real cases that have already been implemented to determine if the results match and, if possible, make corrections. In addition, it is possible to extend this algorithm by adding to possible solutions specific databases and cloud services that provide the desired results to the user in a single database, or a set of them, indicating why such a database was recommended.

Limitations of the study include a short list of questions which can be expanded to enhance the results. In the future this algorithm can be extended with additional questions and databases that can be used. One of main disadvantages of the study is an absence of algorithm testing based on the open data as most of the existing solutions are used in private companies with no access to the data from the outside. The future research directions of correct data storage choice include an idea to add some specific databases considering their advantages and limitations to make the best decision. Another direction is a development of open test environment that can be created to make an algorithm more precise.

4. Conclusions

In this paper, a decision-making algorithm was proposed based on the existing studies analysis. It helps to choose the most efficient way to store large amounts of data based on their characteristics and system requirements. This approach has the potential to be expanded and improved, which can help engineers in the long run at the system design stage.

References

1. Dash, S., Shakyawar, S. K., Sharma, M., Kaushik, S. (2019). Big data in healthcare: management, analysis and future prospects. *Journal of Big Data, 6 (1)*. doi: http://doi.org/10.1186/s40537-019-0217-0
2. Raghupathi, W., Raghupathi, V. (2014). Big data analytics in healthcare: promise and potential. *Health Information Science and Systems, 2 (1)*. doi: http://doi.org/10.1186/2047-2501-2-3
3. Raja, R., Mukherjee, I., Sarkar, B. K. (2020). A Systematic Review of Healthcare Big Data. *Scientific Programming, 2020*, 1–15. doi: http://doi.org/10.1155/2020/5471849
4. Siddiqa, A., Karim, A., Gani, A. (2017). Big data storage technologies: a survey. *Frontiers of Information Technology & Electronic Engineering, 18 (8)*, 1040–1070. doi: http://doi.org/10.1631/fitee.1500441
5. Kumar, S., Singh, M. (2019). Big data analytics for healthcare industry: impact, applications, and tools. *Big Data Mining and Analytics, 2 (1)*, 48–57. doi: http://doi.org/10.26599/bdma.2018.9020031
6. Alonso, S. G., de la Torre Díez, I., Rodrigues, J. J. P. C., Hamrioui, S., López-Coronado, M. (2017). A Systematic Review of Techniques and Sources of Big Data in the Healthcare Sector. *Journal of Medical Systems, 41 (11)*. doi: http://doi.org/10.1007/s10916-017-0832-2
7. Pandey, M. K., Subbiah, K. (2016). A Novel Storage Architecture for Facilitating Efficient Analytics of Health Informatics Big Data in Cloud. *2016 IEEE International Conference on Computer and Information Technology (CIT)*. doi: http://doi.org/10.1109/cit.2016.86
8. Olaronke, I., Oluwaseun, O. (2016). Big data in healthcare: Prospects, challenges and resolutions. *2016 Future Technologies Conference (FTC)*. doi: http://doi.org/10.1109/ftc.2016.7821747
9. Suthakar, U., Magnoni, L., Smith, D. R., Khan, A., Andreeva, J. (2016). An efficient strategy for the collection and storage of large volumes of data for computation. *Journal of Big Data, 3 (1)*. doi: http://doi.org/10.1186/s40537-016-0056-1
10. Geihs, M., Buchmann, J.; Lee, K. (Ed.) (2019). ELSA: Efficient Long-Term Secure Storage of Large Datasets. In: Information Security and Cryptology – ICISC 2018. *ICISC 2018. Lecture Notes in Computer Science*. Cham: Springer, 269–286. doi: https://doi.org/10.1007/978-3-030-12146-4_17

Sofiia Materynska, *Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0002-5746-4899*

✉ ***Vadym Yaremenko,*** *Postgraduate Student, Assistant, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, e-mail: yaremenko.v.s@gmail.com, ORCID: https://orcid.org/0000-0001-8557-6938*

Walery Rogoza, *Doctor of Technical Science, Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0003-2327-156X*

✉ *Corresponding author*