The Evolution of Big Data Analytics Solutions in the Could

Danko Miladinović, Jovan Popović, and Nenad Korolija

Abstract—**Big data analytics is a very important topic both for enterprises, science, and government institutions. The amount of data that is generated is exponentially increasing and the need to analyze data is more important every year. Big data analytics evolved over the past decade from a large on-premises infrastructure for storing and processing data to modern cloud environments. In this paper we discuss how big data analytics evolved over the years and what are the future trends in this area.**

Index Terms—**Big data; cloud; analytics; machine learning; databases.**

I. Introduction

BIG data analytics is one of the most important topics in the IT industry. There is a well-known “Data is the new oil” expression that points out the importance of data analytics that can have a huge effect on modern businesses and economy.

Most of the enterprises either leverage information from their data or plan to extract information from the data they own. Data science and analytics became more important for strategic growth of many organizations.

The organizations and software systems are continuously increasing the amount of data that is generated. Relatively big organizations must face the large amount of data that contains the information important for business decisions. Globally, we are now talking about the Exabyte to Zettabyte scale of data that needs to be processed. The global estimates are that the amount of data to be processed would reach multiple Zettabytes in this decade [1].

In this paper we discuss the industry trends and standards for big data analytics with a focus on data analytics in the cloud. This manuscript describes the solutions offered by the open-source community and the biggest commercial data analytics vendors that pave the way that will be followed by companies. The rest of the document is organized in the following sections:

- In the first section, we will talk about the main problems that impact big data analytic solutions.
- The cloud analytics section describes what are the main benefits of the cloud environments for big data analytics.
- Data analytic solutions section describes the two mainstream approaches for storing and analyzing data: Datawarehouse and Datalakehouse solutions.
- In the data format section, we will discuss the most important aspect of big data analytics – the format that is optimized for storing data.
- The conclusion section summarizes the trends for modern big data analytics.

II. Problem statement

The main problem in big data analytics is the size of data. There are many problems that can be solved by analyzing data stored in files and spreadsheets containing gigabytes of data, or even the relational or NoSQL databases that can contain and process terabytes of data. However, there are many domains where the data contains petabytes of data that cannot be stored in a limited set of files or classic database systems.

The researchers and engineers tried to solve the problem of big data processing using the following approaches:

- Datawarehouses that try to stretch the capabilities of the relational databases by applying distributed processing over large data.
- File-based processing systems that try to build an infrastructure for storing a large amount of data. One of the most widely used solutions is Hadoop with HDFS file system [2].

The big data analytic solutions must ensure that users can store Exabyte scale data and ensure that there is enough compute power to process the data when needed. The infrastructure teams must ensure that they have enough hardware (processors and disk storage) to fulfil the user needs, but at the same time to ensure that the resources are not underutilized or constantly over-provisioned.

Solving the problem of ensuring the required resources for big data analytics, but at the same time not over-provisioning them, appeared to be too hard for the on-premises infrastructure. Planning for the resources could not be both cost effective and ensure enough capacity that will be utilized for most of the time.

Therefore, most of the organizations tried to solve this resource management problem in the cloud making the cloud analytics the mainstream in the big data analytics space.

III. The cloud analytics

Over the past years, the clouds became a very important

Danko Miladinović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: danko@etf.bg.ac.rs).

Jovan Popović is with the Microsoft Research and Development Center, Belgrade. Španskih boraca 3/3, 11000 Belgrade, Serbia (e-mail: jovanpop@microsoft.com).

Nenad Korolija is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: nenadko@etf.bg.ac.rs), (https://orcid.org/1234-1234-1234-123X).

choice for modern big data analytics. There are three main benefits of cloud infrastructure that make them important for big data analytics:

- Large amount of storage that could be used to store any amount of data. The “Data Lake” [3] is a commonly used term for virtually unlimited storage where the organizations might store petabytes of data. As the amount of data rapidly rises, the organizations need to have an infrastructure that will guarantee that they can easily store Exabytes of data, and with the possibility to scale to Zettabytes in the future.
- Large amount of available compute power that could be used for data processing [4]. The compute power needed to analyze data is proportional to the data size and might easily span to the thousands of CPU cores needed to complete the data analytic tasks.
- Cloud resources can be used on-demand and released when they are not needed. This is one of the main reasons for choosing the cloud environment. Most of the data analytic jobs are not continuously processed and might require thousands of CPU cores for data processing, and then suddenly release all the resources. Cloud providers solve this problem using the economy of scale – with the large number of customers there is a high probability that someone will use them once others release them.

We should note that the could environment is not an absolute requirement to have an infrastructure for storing a large amount of data and use thousands of computer cores to process the data. The organizations might use their own data centers, supercomputers, and any custom-built architecture that will organize hundreds or thousands of computers that process the data. This was a common solution for the organizations who built their own Hadoop/HDFS infrastructure [2] for in-house analytics. However, the cost of management includes the need to maintain and replace the hardware, ensure that the infrastructure has enough compute power to satisfy the peak processing, but also to make sure that the capacity is not too underutilized during the period when nobody is executing analytic jobs.

The current trend is that most of the organizations are deciding to delegate the resource management to the cloud providers and utilize the resources on-demand when they need to run some analytics.

The data analytics solutions should not be misguided with the infinite scale claims of the cloud vendors. Cloud environments are built as many computers that are working together to process tasks. The applications see the sum of the compute power, memory, and storage allocated to the computers that are executing the tasks. In many scenarios, this setup can provide “the infinite scale” promise for a variety of applications such as web applications, easily parallelizable functions or jobs, or the classic databases that might not require a constant compute power of up to 128 cores. This kind of infrastructure is the ideal choice for scaling out the large number of small compute units such as microservices, functions that might need to quickly replicate to. These kinds of solutions made of a large number of micro-compute units are perfect for the cloud environments where each unit can be deployed on some available compute node in the cloud [5]. However, these solutions will rarely require an atomic compute unit between 64 and 128 cores. The big data analytics solutions with the demand to store and process petabytes of data might require compute power that can challenge the infinite scale promise of the clouds. In the big data analytics solutions, we can see the impact of the physical infrastructure where the components that process data might require a large amount of CPU or memory needed to decompress the large files and process the information. Therefore, migrating data to cloud and running the analytical functions in the compute provided by the cloud are not enough. In practice, data analytics solutions require specialized services such as cloud Datawarehouse [6] or cloud Datalakehouse [7] solutions, that are able to efficiently combine the physical resources in the cloud and optimally process data.

IV. THE DATA ANALYTIC SOLUTIONS

Data analytics solutions provide infrastructure and tools for the analysts and the business users that enable them to store and analyze data. There are two main classes of data analytic solutions:

- The Datawarehouse solutions that represent centralized data storage with API for analyzing data and implementing the business intelligence solutions [6].
- The Lakehouse solutions represent the analytical solution running on the storage that is detached from the analytical engine [7].

Both classes of the solutions are aware of the underlying infrastructure and designed to optimally process large amounts of data. The main differences between Data Warehouse and Data Lakehouse solutions are given in Table I.

TABLE I
THE KEY DIFFERENCES BETWEEN DATA WAREHOUSE AND DATA LAKEHOUSE

	Warehouse	Lakehouse
Data format	Proprietary and highly optimized.	Based on open specifications.
Data location	Internal – data is ingested from the external source.	External – data is placed on the original locations.
Data access	Through the predefined API or protocol (SQL)	Direct file access using the storage API

The main trade-off between Datawarehouse and Data Lakehouse solutions is the choice between the interactive and the real-time analytics. The Datawarehouses store data in the data format optimized for analytics, which enables them to complete the queries in second-to-minute time span. However, they require data engineers to load data from the actual locations into the Datawarehouse, meaning that the data analysts work with the snapshot of the data taken at the load time. The Lakehouses reference original data in the lakes without a need to ingest the data. However, the raw format of the data cannot guarantee sub-second or even sub-minute performance. Despite the differences, both Datawarehouse and Datalakehouse solutions are used in practice. The following sections describe the main characteristics of these solutions.

A. Warehouse solutions

The traditional database systems containing data used for the analytics are Datawarehouse solutions. Datawarehouse solutions have existed for decades, have well defined techniques for designing Datawarehouse schema [8, 9], and a variety of tools available for advanced data analytics. For a very long time, Datawarehouses were the mainstream solutions for storing and analyzing huge data volumes. The top vendors such as Oracle, Teradata, and Exadata are still enabling enterprises to store and analyze large amounts of data.

The main idea with the Datawarehouse solution is that the data required for analytics must be ingested into the internal data format that is highly optimized for analytics. The advantage of this approach is the performance. Internal and in many cases proprietary format contains optimizations that are not available in the open-source solutions. In the past, the proprietary format gave the Datawarehouse performance advantages compared to other analytic systems.

The main issues in the Datawarehouse solutions are the facts that the underlying infrastructure must enable Datawarehouse to store all required data, which puts a burden on the administration teams, and the cost of the solution that includes the resources that are always allocated to the Datawarehouse system even if it is not used. The Datawarehouse solutions were the first choice for most of the analytic teams who need interactive analytics, but the total cost of ownership (TCO) in many cases does not justify the solution.

Amazon made a breakthrough in the Datawarehousing technology with the release of Redshift [10] – a cloud-native Datawarehouse service that provides full data warehousing experience exposed as a cloud service. The customers got the ability to provision the Datawarehouse service, load the data, and define the compute needed to analyze the data. The biggest advantage of cloud Datawarehouse is the resource elasticity that solved the main drawback of the classical Data warehousing solutions – the TCO. Unlike the on-premises Datawarehouse that had pre-build resources, the cloud Datawarehouse enabled organizations to scale up and down the resources depending on the needs. The organizations could load a large amount of data without worrying about the underlying storage capacities, scale-up the compute power of the Datawarehouse when needed, and scale it down to reduce the cost when there is no need for processing the data. The cloud Datawarehouse provided by Amazon fulfilled the main requirements of the classical warehouses and added elasticity. Other vendors followed this approach and now we have many cloud data warehousing solutions such as Azure Synapse Datawarehouse [11], Google BigQuery [12], Snowflake [13], etc. All vendors are trying to combine all the benefits of the classical Datawarehouse with the elasticity and scale of the cloud.

The modern cloud Datawarehouses solve the problem of elasticity and scale on-demand that cannot be easily solved in the on-premises Datawarehouses. However, there is still one downside – the data must be ingested from the external data sources into the Datawarehouse internal data format, which causes a lag between the latest data and the data available for the analytics.

B. Lakehouse solutions

In cloud environments the data is stored in the Data lakes. The Data lake is a logical storage space where the organizations can store the exabytes of data, get the best throughput for reading raw files, ensure redundancy that can span across multiple geographical regions and data centers. The Data lake seems like a perfect solution for storing data. The only drawback of Data lake is that they do not provide the ability to analyze the data in the lake.

The first successful attempt to provide analytic capabilities over large storage was Hadoop – a distributed system that enabled the analyst to analyze a large amount of data stored in the distributed system called Hadoop file system (HDFS). This setup enabled the analyst to analyze data, but the performance of Hadoop is far from interactive. Apache Spark [14] is one of the most popular platforms that enabled analysts to do efficient analytics on the lake. Apache Spark became mainstream in the data lake solutions. One of the most common query engines used to implement the Lakehouse pattern is Apache Spark. Apache Spark is an open-source distributed query system licensed under Apache License 2.0. Spark enables advanced data analytics, management, and updates, and provides a rich and powerful set of APIs to analyze data.

The main idea of Lakehouse architecture is the separation of compute and storage. The compute is a query engine or data processing engine that is detached from the storage, and the data is placed in Data lake where it can be accessed by any query. The compute engine fetches data from the remote Data lake and return data to the analysts once the processing is completed.

Many commercial vendors offered their own implementation of Data Lakehouse services. Nowadays, we have many Lakehouse-type solutions that are offered on different clouds such by Databricks (proprietary version of Apache Spark code implemented by the founders of Spark), Azure Synapse, etc. The main characteristics of these solutions is that they are always referencing the externally stored data, and don’t require data to be ingested to start analytics. This enables real-time insight into the latest version of data without the need to wait for the daily data loads to finish before starting the analytics.

One of the main concerns in the Lakehouse solution is whether they would be able to match the performance of Warehouse solutions. Traditionally, the proprietary format used in Datawarehouse solutions was the main competitive advantage compared to the original data formats stored in Data lake. Databricks announced that they have set a new world record in 100TB TPC-DS, the gold standard performance benchmark for data warehousing [15]. The test was performed in Barcelona supercomputing center and officially submitted as TPC-DS result that outperformed the previous record by 2.2x. This was the first Lakehouse-class solution that set the record in an official Datawarehouse benchmark and proved that the Lakehouse architectures can compete with the modern Datawarehouse solutions. One of the key reasons for this kind of success of Lakehouse solution is that they are using the optimized storage file format that matches the proprietary internal formats used in the Datawarehouse solutions.

More insights into data lake solutions and current trends can be found in the literature, including what steps are needed to adopt the cloud concept in data analytic solutions [16].

V. Data format

One of the most important design decisions that will impact the efficiency of the data analytics is the choice of the file format that will be used to store the data. Most of the data used for analytic purposes is stored in a plain textual format represented as a delimited text (for example comma separated values – CSV, tab-separated values – TSV, etc.). The documents containing tabular data are represented as Open Office or Microsoft Office formats. Although these data formats are very common, they are inefficient for big data analytics.

There is an additional class of plain textual data represented in JSON format. The JSON format is the standard format in many Internet of Things (IoT) applications where the IoT devices send the messages in JSON format, or the messages that will be eventually stored in JSON format [17]. JSON format provides flexibility for changing the structure of data but complicates the analytics because it requires a parser that is more complex than the plain delimited text parser.

In the practice, the data analysts could analyze data stored in CSV, JSON and other commonly used formats. However, since these formats are not optimized for analytics, it was very hard for data analysts to extract valuable information with performance that matches performance of database systems.

The proprietary data formats that are used to store data in Datawarehouse solutions were the biggest competitive advantage of the Datawarehouse systems compared to the open-source solutions. The optimized formats with high compression, columnar organization of data, and vectorized processing was the main reason why the data analytics teams used the Datawarehouse solutions.

In the open-source community multiple advanced formats are proposed that are designed to optimize the storage format and improve the performance of analytics. Examples are Row Columnar (RC)[18] or Optimized Row Columnar (ORC)[19] file format. The idea of these formats was to define binary representation of data prepared for analytics and optimize access for the analytical jobs. However, the open-source format that took most of the market share became Parquet format [19]. Parquet format is an open-source format that introduces most of the benefits that exist in the proprietary Datawarehouse formats, such as:

- Column organization – the data is physically separated into column segments instead of rows. The column segments contain all cell values from the same column. The columnar organization enables the analytical queries that read 2 columns out of 100 columns to read only 2% of data on average. Since the analytical queries aggregate the measures and summarize them by few columns, the columnar organization introduces most of the performance benefits for the analytical queries.
- Row-groups – columns are divided into row–groups (for example 100.000 rows represent a row group that will be split into the columns) The column segments within the row groups contain some statistical information about cells such as min/max values.
- Compression – There are some compression techniques such as run-length encodings (RLE) [20] that can be applied on the Parquet files to achieve excellent 10-100x compression, which matches the compression in the proprietary Datawarehouse formats. The main impact is not just storage savings. Compressed storage decreases IO requests sent to the storage and improves data throughput that analytical tools can use to fetch the data.
- Non-relational types – Parquet is not limited to strongly defined types and enables storing objects and arrays. The organization of complex types in Parquet format is described in [21].

The Parquet format became the mainstream in data analytics. Although there are other formats that are used in practice, the Parquet format is getting the highest market share and we can expect that the majority of data will be stored in the Parquet format. Therefore, any modern big analytical solution must be based on the Parquet format, or some enhancement based on Parquet. Even if the new file format arrives in the future, there is a high chance that most data will be stored in the Parquet format.

Although the Parquet format is designed to store analytical data that should be read-only (or append-only), there is a need to enable data engineers to make updates to the data. There are several updateable formats (such as Delta Lake [22]), that combine the excellent storage format for analytics and provide ACID guarantees of the operations that managed data. Another possible advancement includes parsing big data using the dataflow paradigm [23] by transforming automatically the parsing software [24] and using appropriate scheduling techniques [25] for the dataflow supercomputing architecture.

VI. Conclusion

Big data analytics solutions evolved from the original on-premises based big Datawarehouse solutions to modern cloud based Datawarehouse solutions. The original on-premises Datawarehouse solutions enabled organizations to store large amounts of data and analyze data with acceptable performance. However, these kinds of solutions failed to enable scalability and elasticity. Cloud computing can scale resources on-demand. Data lake that can store Exabytes of data with guaranteed replication, and advances in the opensource file formats disrupted the Datawarehouse solutions in big data analytics. Although Datawarehouse solutions evolved and have been adapted for the modern cloud environments, architectures with full separation of compute power and storage, where the compute can scale when needed, have a direct access to the latest version of data in the lake, and the performance that match modern Datawarehouses. In addition to matching all features that were historically considered as the advantage of Datawarehouse, the Lakehouse solves the issue that fundamentally cannot be solved with Datawarehouse solutions – data ingestion. Lakehouse solutions are able to access the original data in the Data lake and don’t require an explicit process to load external data. Without performance degradation compared to internal data formats used in Datawarehouse solutions, direct access simplifies data management process by avoiding the additional processes that constantly move the data and also enable analyst to get the data without any delay.

By looking at the modern trends, we can conclude that the future of big data analytics will be based on the cloud environments and Lakehouse architectures. The cloud Data Lakehouse solutions leverage all benefits of the cloud and match performance of the Datawarehouse solutions. The cloud Data Lakehouse solutions can be considered as a primary solution for most of the future research and as the mainstream and preferred technology for development projects.

Acknowledgment

DM and NK are partially supported by the School of Electrical Engineering, University of Belgrade, Serbia and by the Ministry of Education, Science, and Technological Development of the Republic of Serbia. NK is partially supported by the Institute of Physics Belgrade, contract no. 0801-1264/1.

References

[1] P. Chauhan, M. Sood, “Big Data: Present and Future,” The IEEE Computer Society, (2021), DOI: 10.1109/MC.2021.3057442.

[2] S. G. Manikandan, S. Ravi, "Big Data Analysis Using Apache Hadoop," International Conference on IT Convergence and Security (ICITCS), pp. 1-4, (2014). doi: 10.1109/ICITCS.2014.7021746.

[3] E. Zagan, M. Danubianu, "Cloud DATA LAKE: The new trend of data storage," 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, 1-4 (2021), doi: 10.1109/HORA52670.2021.9461293.

[4] I. Hashem, I. Yaqoob, N. Anuar, S. Mokhtar, A. Gani, A., and S. Khan, “The rise of “big data” on cloud computing: Review and open research issues”, Information Systems, Volume 47, 2015, Pages 98-115, ISSN 0306-4379.

[5] R. Han, L. Guo, M. M. Ghanem and Y. Guo, "Lightweight Resource Scaling for Cloud Applications," 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 644-651, (2012). doi: 10.1109/CCGrid.2012.52.

[6] G. Garani, A. Chernov, I. Savvas and M. Butakova, "A Data Warehouse Approach for Business Intelligence," 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, 70-75 (2019). doi: 10.1109/WETICE.2019.00022.

[7] D. Oreščanin and T. Hlupić, "Data Lakehouse - a Novel Step in Analytics Architecture," 44th International Convention on Information, Communication and Electronic Technology, 1242-1246 (2021). doi: 10.23919/MIPRO52101.2021.9597091.

[8] R. Kimball, M. Ross, “The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling,” 2nd. edition. John Wiley & Sons, Inc., USA, (2002).

[9] W. Inmon, “Building the Data Warehouse,” John Wiley & Sons, Inc., USA, (1992).

[10] A. Gupta, D. Agarwal, D. Tan, J. Kulesza, R. Pathak, S. Stefani, and V. Srinivasan, “Amazon Redshift and the case for simpler data warehouses,” SIGMOD (2015).

[11] J. Aguilar-Saborit, R. Ramakrishnan, K. Srinivasan, K. Bocksrocker, I. Alagiannis, M. Sankara, M. Shafiei, J. Blakeley, G. Dasarathy, S. Dash, L. Davidovic, M. Damjanic, S. Djunic, N. Djurkic, C. Feddersen, C. Galindo-Legaria, A. Halverson, M. Kovacevic, N. Kicovic, G. Lukic, D. Maksimovic, A. Manic, N. Markovic, B. Mihic, U. Milic, M. Milojevic, T. Nayak, M. Potocnik, M. Radic, B. Radivojevic, S. Rangarajan, M. Ruzic, M. Simic, M. Sosic, I. Stanko, M. Stikic, S. Stanojkov, V. Stefanovic, M. Sukovic, A. Tomic, D. Tomic, S. Toscano, D. Trifunovic, V. Vasic, T. Verona, A. Vujic, N. Vujic, M. Vukovic, M. Zivanovic, “POLARIS: The distributed SQL engine in Azure Synapse” PVLDB, vol. 13, issue 12, (2020).

[12] K. Sato, “An inside look at Google BigQuery,” Technical report, Google. https://cloud.google.com/files/BigQueryTechnicalWP.pdf.

[13] B. Dageville, T. Cruanes, M. Zukowski, V. Antonov, A. Avanes, J. Bock, J. Claybaugh, D. Engovatov, M. Hentschel, J. Huang, A. W. Lee, A. Motivala, A. Q. Munir, S. Pelley, P. Povinec, G. Rahn, S. Triantafyllis, P. Unterbrunner, “The Snowflake Elastic Data Warehouse,” SIGMOD (2016).

[14] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, M. Zaharia, “Spark SQL: Relational data processing in Spark,” Armbrust, Melbourne, Victoria, Australia, ACM, SIGMOD, (2015).

[15] R. Xin, M. Mokhtar, "Databricks Sets Official Data Warehousing Performance Record," November, 2021, Databricks company blog.

[16] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, ”Leveraging the data lake: Current state and challenges,” International Conference on Big Data Analytics and Knowledge Discovery, Springer, Cham, 179-188, (2019, August).

[17] N. Nikolov, "Research of the Communication Protocols between the IoT Embedded System and the Cloud Structure," 2018 IEEE XXVII International Scientific Conference Electronics - ET, 1-4 (2018). doi: 10.1109/ET.2018.8549604.

[18] Y. He, R. Lee, Y. Huai, Z. Shao, N. Jain, X. Zhang, and Z. Xu, “RCFile: A fast and space-efficient data placement structure in MapReduce-based warehouse systems,” Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, IEEE Computer Society, USA, 1199–1208, (2011). DOI:https://doi.org/10.1109/ICDE.2011.5767933

[19] T. Ivanov, M. Pergolesi, “The impact of columnar file formats on SQL-on-Hadoop engine performance: A study on ORC and Parquet. Concurrency,” Computat Pract Exper. (2020), https://doi.org/10.1002/cpe.5523.

[20] A. Ishtiaq, S. Ahmad, and D. S. Shukla, “Fast Retrieval with Column Store using RLE Compression Algorithm,” International Journal of Computer Applications, 111. 30-34, (2015). 10.5120/19537-1193.

[21] S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis, “Dremel: Interactive Analysis of Web-Scale Datasets”, Proc. of the 36th Int'l Conf on Very Large Data Bases, 330-339 (2010).

[22] M. Armbrust, T. Das, L. Sun, B. Yavuz, S. Zhu, M. Murthy, J. Torres, H. van Hovell, A. Ionescu, A. Łuszczak, M. Świtakowski, M.

Szafrański, X. Li, T. Ueshin, M. Mokhtar, P. Boncz, A. Ghodsi, S. Paranjpye, P. Senster, R. Xin, and M. Zaharia, "Delta lake: high-performance ACID table storage over cloud object stores," Proceedings of the VLDB Endowment, vol. 13, issue 12, 3411–3424 (2020) DOI:https://doi.org/10.14778/3415478.3415560.

[23] N. Korolija, J. Popović, M. Cvetanović, and M. Bojović, "Dataflow-based parallelization of control-flow algorithms,“ *Advances in computers*, Elsevier, **104**, 73-124 (2017).

[24] V. Milutinovic, J. Salom, D. Veljovic, N. Korolija, D. Markovic, and L. Petrovic, "Transforming applications from the control flow to the dataflow paradigm," *Dataflow supercomputing essentials*, Springer, Cham, 107-129 (2017).

[25] N. Korolija, D. Bojić, A. R. Hurson, and V. Milutinovic, "A runtime job scheduling algorithm for cluster architectures with dataflow accelerators," *Advances in computers*, Elsevier, **126** (2022).