applied sciences

MDPI

Article

Experimental Characteristics Study of Data Storage Formats for Data Marts Development within Data Lakes

Vladimir Belov [1], Alexander N. Kosenkov [2] and Evgeny Nikulchev [1,*]

[1] Department of Intelligent Information Security Systems, MIREA—Russian Technological University, 119454 Moscow, Russia; belov_v.a@mail.ru
[2] Department of Hospital Surgery, Sechenov First Moscow State Medical University, 119435 Moscow, Russia; alenkos@rambler.ru
* Correspondence: nikulchev@mail.ru

Abstract: One of the most popular methods for building analytical platforms involves the use of the concept of data lakes. A data lake is a storage system in which the data are presented in their original format, making it difficult to conduct analytics or present aggregated data. To solve this issue, data marts are used, representing environments of stored data of highly specialized information, focused on the requests of employees of a certain department, the vector of an organization's work. This article presents a study of big data storage formats in the Apache Hadoop platform when used to build data marts.

Keywords: big data; data lakes; data storage formats; data marts

Citation: Belov, V.; Kosenkov, A.N.; Nikulchev, E. Experimental Characteristics Study of Data Storage Formats for Data Marts Development within Data Lakes. *Appl. Sci.* **2021**, *11*, 8651. https://doi.org/10.3390/app11188651

Academic Editor: Luis Javier Garcia Villalba

Received: 25 August 2021
Accepted: 15 September 2021
Published: 17 September 2021

1. Introduction

When developing analytical systems, solving the issue of storing the loaded data becomes an important task. One of the most popular methods for storing information involves the use of relational databases [1]. However, the growth in the volume of data, the increase in the number of data acquisition channels, as well as the diversity of the structures of the data obtained have led to the emergence of several directions of research in the field of data storage [2]. Among such areas, the concept of big data stands out [3]. The concept of big data is based on six aspects: value, volume, speed, variety, reliability, and variability [4]. This means that the term "big data" refers not only to the volume of these data, but also their ability to act as sources for generating valuable information and ideas [4].

Among the modern methods of storing big data, the architectural concept of building data lakes stands out [5]. The data lake is a scalable system for storing and analyzing data stored in their original format and used to extract knowledge [5]. It is clear from the definition that information enters such a system with an original structure, which requires the creation of a tool for reducing data to a single structure that is understandable to the end user. In addition, when analyzing information it is often necessary to select only highly specialized data intended for solving a specific problem [6].

To solve the described problems, the technology of creating data marts is used [7]. Data marts are a slice of the stored information with a highly specialized focus on the requests of the employees of a particular department or the specifics of the tasks assigned.

One of the software tools that allows the building of both data lakes and data marts is the Apache Hadoop platform [8]. Data storage in this system is supported by writing information to files in the HDFS (Hadoop Distributed File System) [9]. As the HDFS does not have specialized file formats, it becomes necessary to choose a storage format from a variety of developed formats. Among the most widely known formats used in Apache Hadoop and the HDFS are JSON [10], CSV [11], Apache Parquet [12], Apache Avro [13], and Apache ORC [14]. Each of these file formats has its own characteristics in the file

structure, which determines its characteristics in terms of such criteria as write speed, read speed, occupied space, and others.

This paper presents a study of data storage formats in relation to the creation of data marts for the tasks of creating reports during mass testing. The data source for the study was the Digital Psychological Platform for Mass Web-Surveys [15], developed for conducting psychological research in educational institutions using the methods of mass online testing.

The aim of this paper is to study the use of different HDFS file formats applied to data mart development within the Apache Hadoop platform and to assess the effectiveness of the most popular file formats used in the HDFS. The study analyzed the effectiveness of choosing a format for building reports based on the data obtained. To analyze the formats, a number of criteria were selected, and experiments were carried out on the Hadoop clusters.

This paper consists of six sections. The second section provides an analysis of works related to the topic of this research. It provides an overview of the main research studies aimed at exploring various storage media in large systems and the main trends, and it justifies the need for ongoing research. The third section provides an overview of the studied data storage formats in the HDFS, the internal structure of data formats, supported data types, and data storage methods. The fourth section describes the experiment carried out and gives information about the configuration of the experimental stand, information about the data used for the experiment, and the experimental scheme. The fifth section presents the results of the experiment, derived by assessing the time and resource efficiency of these formats in relation to the creation of data marts. The sixth section describes the findings of this study.

2. Related Works

Studying the techniques of data storing is one of the most important challenges, not only for commercial software development, but also in the field of scientific research. There are many studies aimed at finding the most optimal tools for storing data, as well as methods of storing data, depending on the software architectures used [16–29]. Depending on the approach to data storage methods, several groups of studies can be distinguished.

Most of the research in the field of data storage is related to the analysis of NoSQL solutions [16], as well as the choice between relational and NoSQL databases [17]. Of the greatest interest in the preparation of this article is the research related to the use of relational and NoSQL databases to create data marts. The study in [18] describes an approach to creating a data mart based on a MongoDB document database, which implies the use of the JSON format for the final selection. In turn, relational databases have wider functionality, both for creating data marts and as tools for OLAP technologies [19].

Exploring different storage file formats represents a different area of research. In [20–24], studies of big data storage formats, such as Apache Avro, Apache Parquet, ORC, and others are presented. These studies present the results of studying various formats in terms of performance or format selection for a specific purpose. Thus, the research in [20] is related to the study of file formats for storing data in web systems. The most popular Apache Parquet and Apache Avro formats are explored here. The research in [21] aims to study data storage formats for problems in the field of bioinformatics and examines formats, such as Apache Parquet, ORC, Apache Avro, and other popular formats. In [22], similar problems of resource efficiency are considered, but only the storage formats of Apache Avro and Apache Parquet are investigated. The research in [23] is a comprehensive study of the ORC and Apache Parquet file formats, including load testing using various tools for working with these formats. These formats are very similar in structure and data storage methods, making this research the most valuable. However, this study is more theoretical in nature as it aims to study the formats themselves without being tied to a specific task.

Despite the detailed study of data storage technologies, the study of data marts using the Apache Hadoop platform is a novel direction. Research aimed at exploring these technologies is often either conceptual-level research or considers Apache Hadoop only

as a source of data for creating data marts in other systems. For example, [24] describes a conceptual model for creating data marts in MPP Greenplum using data from a data lake based on Apache Hadoop.

This study supplements the analyzed studies and considers a number of the most popular formats in relation to the tasks of developing data marts within the Apache Hadoop platform without using additional data storage tools.

In addition, there are a number of studies related to developments in the field of software integration and the choice of the optimal tool for various purposes using different decision-making methods [25–28]. The choice of system components is an important condition for the correct operation of the software complex. As data storage files are part of the analytical platform, studying the possibilities of various formats for developing data marts is an important task for the correct operation of the system as a whole.

3. Storage Formats Overview

One of the tools for working with data is the so-called data mart. Data marts refer to thematic data environments that address analytics in specific areas.

When working with data using data lakes, the creation of data marts is necessary because reading all the contained data is time-consuming as well as resource-intensive. The development of data marts facilitates further work with data.

When developing data lakes based on the Apache Hadoop platform, the platform defines the task of choosing a data storage format. The HDFS does not have a default format. Depending on the task, a choice of data storage format is required.

The most popular formats for storing data in the HDFS are JSON, CSV, Apache Parquet, Apache Avro, and Apache ORC. The main features of these formats will be discussed below.

Avro is a line-oriented data storage format. The main feature of the format is the presence of a schema in the JSON format, which allows faster reading and interpretation operations [13]. The file structure consists of a header and data blocks. In addition, this format supports the evolution of data schemas, handling schema changes by omitting, adding, or modifying individual fields. Avro is not a strongly typed format: the type of information for each field is stored in the metadata section along with the schema. Due to this, prior knowledge of the schema is not required to read the serialized information [13].

Comma-Separated Values is a text format that describes data in tabular form. The structure of the CSV file is represented as strings separated by commas. A title is assumed, but this is not a strict requirement. The CSV file does not support different types and data structures—all data are presented as strings.

JavaScript Object Notation is a simple text format based on a subset of the JavaScript programming language. This format is most widely used in data exchange systems, API development, and remote procedure calls [10]. However, with the advent of NoSQL solutions, the JSON format has gained popularity in storing big data in document databases [29]. JSON supports data types and structures, such as string, number, Boolean, arrays, null, and internal objects [10].

Optimized Row Columnar is a column-oriented (columnar) storage format for big data in the Apache Hadoop ecosystem [14]. ORC is optimized for reading big data streams, including integrated support for quickly finding the required rows. Columnar storage allows one to read, unpack, and process only those values that are needed for the current request. As data in ORC are strongly typed, an encoding is therefore chosen when writing that is the most suitable for each datum type, creating an internal index as the file is written [14].

Apache Parquet is a column-oriented binary format that takes advantage of the concise presentation of information. Parquet allows one to specify compression schemes at the column level and add new encodings as they appear [12]. Parquet uses an architecture based on “definition levels” and “repetition levels”, which allows data encoding quite efficiently, and information about the schema is put into separate metadata [12]. The

Parquet format explicitly separates metadata from data, allowing columns to be split across multiple files, as well as having a single metadata file referencing multiple Parquet files.

4. Experiment

For the experimental evaluation, an Apache Hadoop cluster was used. It has the characteristics described in Table 1.

Table 1. Experimental stand configuration.

Configuration Item	Characteristics
Nodes count	3
Replication factor	2
Hadoop Distributive	Cloudera v. 4.2
Nodes Characteristics	
Processor	Intel Core i7 2 Cores
RAM	4 GB
OS	Debian 9
Platform	Java Virtual Machine
Programming language	Java v.1.8
Framework	Apache Spark v. 2.4

For the study, a dataset obtained using the Digital Psychological Platform for Mass Web-Surveys [15] was used. This platform collects data using web interfaces built into the platform itself. Based on the surveys, a population data base is formed. These data become available for analysis by interdisciplinary research teams.

The data are a set of JSON objects describing the results of the online testing of each individual participant. Figure 1 shows a simplified example of such an object.

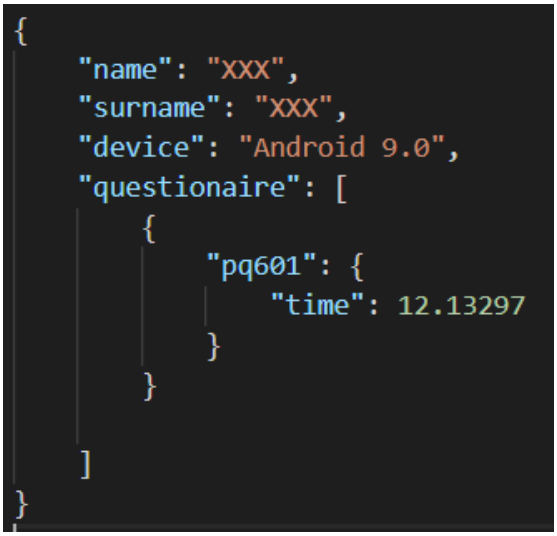

Figure 1. An example of an object containing information about the passing of testing by one participant.

Each JSON object contains the following information about the completed questionnaire:

- name and surname of the test taker;
- the device platform from which the test was passed (commonly, it is information about the operating system installed on the device)—this information can help in understanding which type of device (PC, mobile, old platforms) is used to improve the support of such operating systems;
- an array of test completion information for each question. The most interesting information for this study from the array is the time the test taker spent on each question.

As a result, it is necessary to obtain a data mart containing aggregated information for each question and device: the type of device, question number, mean response time, median time, standard deviation, and first and third quartiles. The total volume of all the data prepared for the experiment is 2 GB. These data were obtained during mass psychological testing in different schools of the country.

An application was developed using the Apache Spark framework [30], which has a large set of functions for working with data in a distributed system. The application implements an algorithm for reading, creating a data mart, and writing to a specific format, depending on the transmitted parameters.

Figures 2–4 show the implemented algorithms for creating a data mart based on the resulting dataset.

```
private static Dataset<Row> readData(SparkSession spark) {
    Dataset<Row> df = spark.read().json(PATH);
    df = df.drop(col("_corrupt_record"))
            .na().drop("all")
            .dropDuplicates();
    return df;
}
```

Figure 2. Reading and clearing data.

```
private static Dataset<Row> createDataMart(Dataset<Row> df) {
    return df
            .withColumn("deviceType",
                    callUDF("deviceTypeFunc", col("device")))
            .groupBy(col("qNum"), col("device"))
            .avg("qTime")
            .withColumn("meanTime",
                    callUDF("meanFunc", col("qTime")))
            .withColumn("medianTime",
                    callUDF("medianFunc", col("qTime")))
            .withColumn("standardDev",
                    callUDF("standardDevFunc", col("qTime")))
            .withColumn("firstQuartile",
                    callUDF("firstQuartileFunc", col("qTime")))
            .withColumn("thirdQuartile",
                    callUDF("thirdQuartileFunc", col("qTime")))
            .select("qNum",
                    "deviceType",
                    "meanTime",
                    "medianTime",
                    "standardDev",
                    "firstQuartile",
                    "thirdQuartile");
}
```

Figure 3. Developing data mart.

The experiment was as follows. From the dataset, it is necessary to prepare a mart containing statistical data on all the participants in the mass testing. Test pass data are a set of JSON files containing JSON objects for each participant in a bulk test. All the files are located in the HDFS cluster. The developed application was used to build the showcase.

```
private static void write(Dataset<Row> df) {
    df.write().format(FORMAT).save(WRITE_PATH);
}
```

Figure 4. Writing data.

The experiment is shown in Figure 5. The spark-submit function launches a Spark application and reads data from the HDFS. To build a report, a data frame is generated, consisting of data aggregated by the application. The result is written into one of the data storage formats. Then, a link to the location is automatically created to create a table in Apache Hive [31], which is necessary for further reading of the results by employees without programming skills, as well as by analysts. From the moment the application starts working until its termination, the time of working with the data is measured.

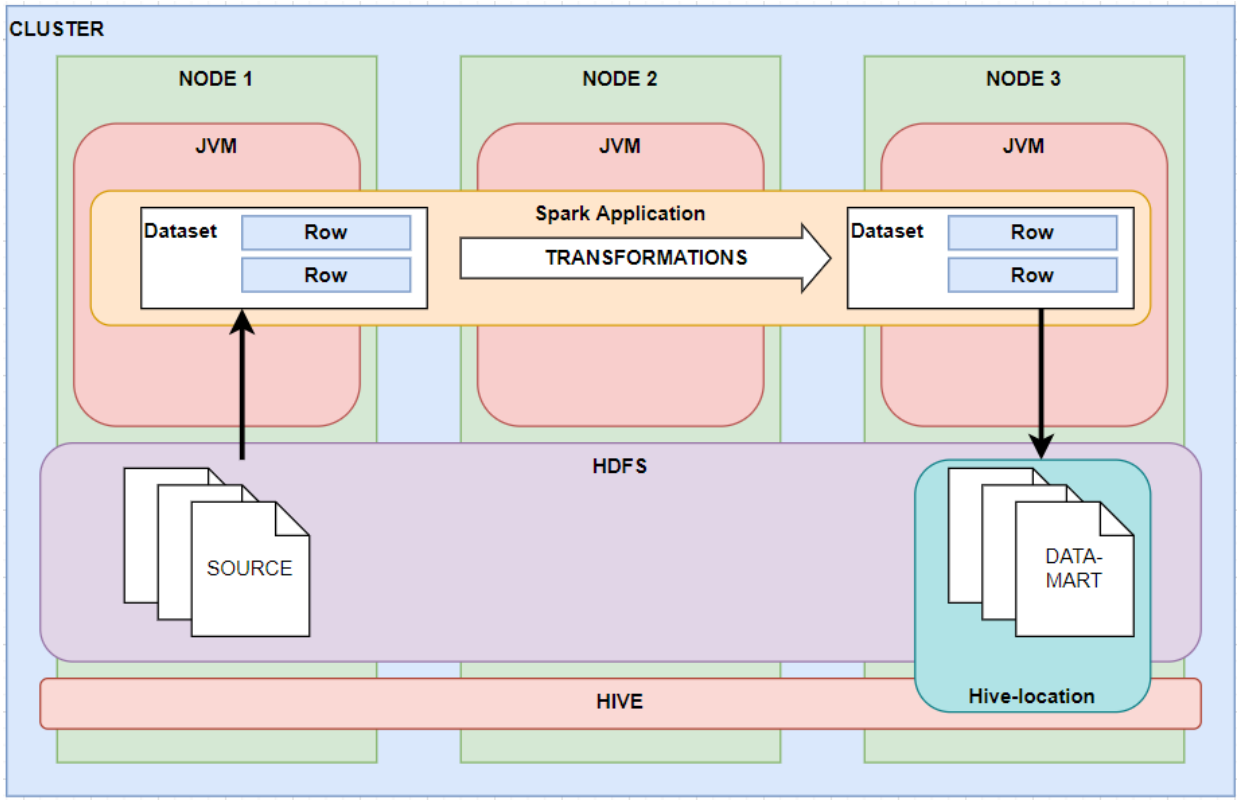

Figure 5. Schema of the experiment.

5. Results

The results of the total read and write time obtained are presented in Figure 6.

The results of the experiment show that the Apache Parquet format has the fastest result.

In addition, previous studies [32] show that this format has the fastest data reading, which leads to the conclusion that it is advisable to use this format. Another format that showed excellent results was the ORC format. However, this format requires additional improvements to be able to work in Hive. Other formats showed the worst results, and therefore their use is not considered appropriate.

The result of estimating the data volume is shown in Figure 7.

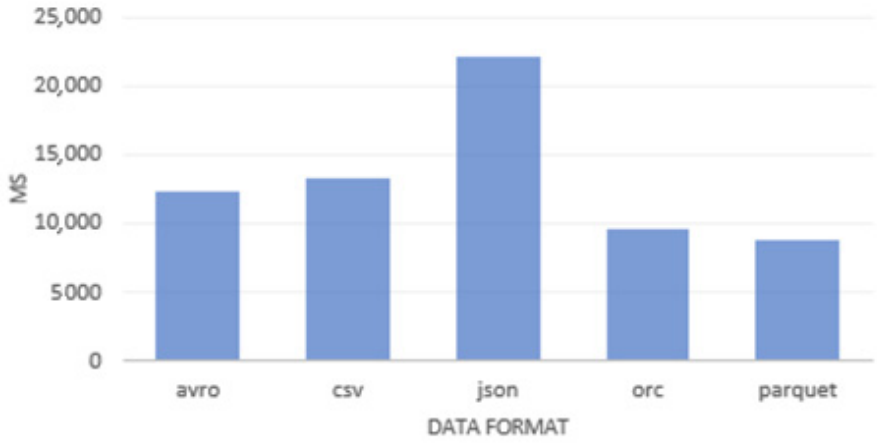

Figure 6. Time assessment of storage formats.

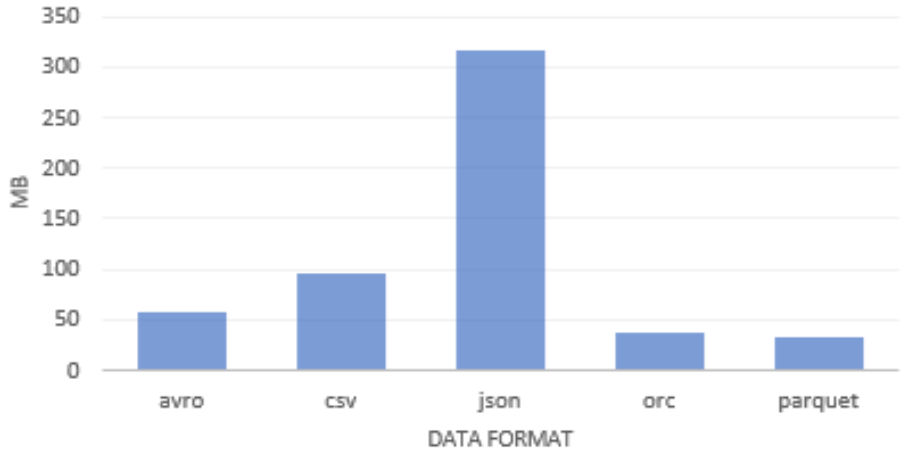

Figure 7. Resulting volume of data.

6. Conclusions

This article presented a study of various data storage formats supported by the Apache Hadoop system for building data marts in relation to the tasks of creating reports based on the results of mass online testing.

As the main performance criteria, we selected such characteristics as the time it takes to create a data mart, as well as its volume.

The study consisted of a series of experimental launches of the developed application using the Apache Spark framework, which has a large set of functions for working with data in the Apache Hadoop distributed system.

Consequently, the results obtained showed that it is most appropriate to use the Apache Parquet format in this case. Other formats showed the worst results.

This study is part of a larger study related to the development of an analytical platform for educational organizations. The results of this study will be applied to the next studies related to the application of various data storage formats in the Apache Hadoop platform. In addition, it is planned to conduct a study comparing the use of the Apache Hadoop platform and MPP architectures such as Greenplum for the development of data marts.

Author Contributions: Conceptualization E.N., A.N.K., V.B.; methodology, V.B.; software, V.B.; validation, E.N., A.N.K.; data curation, E.N., A.N.K., V.B.; writing—original draft preparation, V.B.; writing—review and editing, E.N., A.N.K.; visualization, V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alasta, A.F.; Enaba, M.A. Data warehouse on Manpower Employment for Decision Support System. *Int. J. Comput. Commun. Instrum. Eng.* **2014**, *1*, 48–53.
2. Chong, D.; Shi, H. Big data analytics: A literature review. *J. Manag. Anal.* **2015**, *2*, 175–201. [CrossRef]
3. Yang, C.; Huang, Q.; Li, Z.; Liu, K.; Hu, F. Big Data and cloud computing: Innovation opportunities and challenges. *Int. J. Digit. Earth* **2017**, *10*, 13–53. [CrossRef]
4. Cappa, F.; Oriani, R.; Peruffo, E.; McCarthy, I.P. Big Data for Creating and Capturing Value in the Digitalized Environment: Unpacking the Effects of Volume, Variety and Veracity on Firm Performance. *J. Prod. Innov. Manag.* **2020**. [CrossRef]
5. Khine, P.P.; Wang, Z.S. Data Lake: A new ideology in big data era. *ITM Web Conf.* **2018**, *17*. [CrossRef]
6. Tomashevskaya, V.S.; Yakovlev, D.A. Research of unstructured data interpretation problems. *Russ. Technol. J.* **2021**, *9*, 7–17. [CrossRef]
7. Ghezzi, C. Designing data marts for data warehouses. *ACM Trans. Softw. Eng. Methodol.* **2001**, *10*, 452–483. [CrossRef]
8. O'Driscoll, A.; Belogrudov, V.; Carroll, J.; Kropp, K.; Walsh, P.; Ghazal, P.; Sleator, R.D. HBLAST: Parallelised sequence Similarity—A Hadoop MapReducable basic local alignment search tool. *J. Biomed. Inf.* **2015**, *54*, 58–64. [CrossRef]
9. HDFS. 2020 HDFS Architecture Guide. Available online: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html (accessed on 24 July 2021).
10. Introducing JSON. Available online: https://www.json.org/json-en.html (accessed on 22 August 2021).
11. Super CSV. What is CSV? Available online: http://super-csv.github.io/super-csv/csv_specification.html (accessed on 22 August 2021).
12. Apache. Parquet Official Documentation 2018. Available online: https://parquet.apache.org/documen-tation/latest/ (accessed on 24 July 2021).
13. Apache. Avro Specification 2012. Available online: http://avro.apache.org/docs/current/spec.html (accessed on 24 July 2021).
14. ORC. ORC Specification 2020. Available online: https://orc.apache.org/specification/ORCv1/ (accessed on 24 July 2021).
15. Nikulchev, E.; Ilin, D.; Silaeva, A.; Kolyasnikov, P.; Belov, V.; Runtov, A.; Pushkin, P.; Laptev, N.; Alexeenko, A.; Magomedov, S.; et al. Digital Psychological Platform for Mass Web-Surveys. *Data* **2020**, *5*, 95. [CrossRef]
16. Rasheed, Y.; Qutqut, M.H.; Almasalha, F. Overview of the Current Status of NoSQL Database. *Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 47–53.
17. Ali, W.; Shafique, M.U.; Majeed, M.A.; Raza, A. Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. *Asian J. Res. Comput. Sci.* **2019**, *4*, 1–10. [CrossRef]
18. Bicevska, Z.; Oditis, I. Towards NoSQL-based Data Warehouse Solutions. *Procedia Comput. Sci.* **2017**, *104*, 104–111. [CrossRef]
19. Hamoud, A.K.; Ulkareem, M.A.; Hussain, H.N.; Mohammed, Z.A.; Salih, G.M. Improve HR Decision-Making Based On Data Mart and OLAP. *J. Phys. Conf. Ser.* **2020**, *1530*, 012058. [CrossRef]
20. Wang, X.; Xie, Z. The Case for Alternative Web Archival Formats to Expedite the Data-To-Insight Cycle. In Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, Virtual Event, China, 1–5 August 2020; pp. 177–186.
21. Ahmed, S.; Ali, M.U.; Ferzund, J.; Sarwar, M.A.; Rehman, A.; Mehmood, A. Modern Data Formats for Big Bioinformatics Data Analytics. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*. [CrossRef]
22. Ramírez, A.; Parejo, J.A.; Romero, J.R.; Segura, S.; Ruiz-Cortés, A. Evolutionary composition of QoS-aware web services: A many-objective perspective. *Expert Syst. Appl.* **2017**, *72*, 357–370. [CrossRef]
23. Plase, D.; Niedrite, L.; Taranovs, R. A Comparison of HDFS Compact Data Formats: Avro Versus Parquet. *Moksl. Liet. Ateitis* **2017**, *9*, 267–276. [CrossRef]
24. Raevich, A.; Dobronets, B.; Popova, O.; Raevich, K. Conceptual model of operational-analytical data marts for big data processing. *E3S Web Conf* **2020**, *149*. [CrossRef]
25. McCarthy, S. Reusing Dynamic Data Marts for Query Management in an on-Demand ETL Architecture. Ph.D. Thesis, Dublin City University, Dublin, Ireland, 2021.
26. Huh, J.H.; Seo, K. Design and test bed experiments of server operation system using virtualization technology. *Hum. Cent. Comput. Inf. Sci.* **2016**, *6*, 1. [CrossRef]
27. Yang, Q.; Ge, M.; Helfert, M. Developing Reliable Taxonomic Features for Data Warehouse Architectures. In Proceedings of the IEEE 22nd Conference on Business Informatics (CBI), Antwerp, Belgium, 22–24 June 2020; pp. 241–249. [CrossRef]
28. Nikulchev, E.; Ilin, D.; Gusev, A. Technology Stack Selection Model for Software Design of Digital Platforms. *Mathematics* **2021**, *9*, 308. [CrossRef]
29. Oussous, A.; Benjelloun, F.-Z.; Lahcen, A.A.; Belfkih, S. NoSQL databases for big data. *Int. J. Big Data Intell.* **2017**, *4*, 171–185. [CrossRef]

30. Salloum, S.; Dautov, R.; Chen, X.; Peng, P.X.; Huang, J.Z. Big data analytics on Apache Spark. *Int. J. Data Sci. Anal.* **2016**, *1*, 145–164. [CrossRef]
31. Apache. Hive Official Documentation 2014. Available online: https://hive.apache.org/ (accessed on 24 July 2021).
32. Belov, V.; Tatarintsev, A.; Nikulchev, E. Choosing a Data Storage Format in the Apache Hadoop System Based on Experimental Evaluation Using Apache Spark. *Symmetry* **2021**, *13*, 195. [CrossRef]