Self-Adaptable Infrastructure Management for Analyzing the Efficiency of Big Data Stores

Konstantinos Mavrogiorgos, Athanasios Kiourtis, Argyro Mavrogiorgou, and Dimosthenis Kyriazis
Department of Digital Systems, University of Piraeus, Piraeus, Greece
Email: {komav, kiourtis, margy, dimos}@unipi.gr

***Abstract*—Currently a continuously increasing amount of data is generated and processed in a daily basis towards improving decision-making and facilitating the gaining of insights. In this context, current era is characterized as the "Era of Big Data" with data characteristics including high volume, velocity, variety, or veracity, creating multiple chances and challenges. Several Information and Communication Technology (ICT) firms, enterprises and research projects are working upon the overall Big Data challenges with an increasing amount of effort being given to identify the means of effectively and efficiently collecting, storing, retrieving, analyzing and reusing Big Data in order to improve their services, increase their competitive advantage and support competent decisions. Such approaches deal with several sectors including the domains of healthcare, agricultural, environmental, transportation, governance, or insurance. Towards this goal, in order to identify the most efficient and less-time consuming database for using and reusing the stored data, in this paper we contribute into the selection of the most appropriate database for efficiently storing and retrieving Big Data. More specifically, considering the challenges and the nature of Big Data, as well as the main categories of databases that currently exist, three (3) NoSQL document-based databases are being described and compared under different working environments and conditions, namely the ArangoDB, the MongoDB and the CouchDB. These working environments depend on the Diastema platform that provides the ability of the adaptive allocation and management of infrastructures based on the networking, computing, and storing requirements of each database. Consequently, the overall performance and efficiency of these databases is calculated along with the latter platform, and is being based on specific metrics and criteria, which include the average execution time of CRUD operations and the corresponding requirements for resources, thus concluding to the most suitable databases to store Big Data.**

***Index Terms*—big data, storage, document-based, infrastructure management, ArangoDB, MongoDB, CouchDB**

I. Introduction

Currently, there exists a tremendous amount of data that is either generated and/or processed. For instance, it is estimated that the pieces of content uploaded to Facebook are just about thirty (30) billion while the value of big data for the healthcare industry is about 300 billion [1]. It is also worth mentioning that the Big Data analytics market revenue worldwide is expected to reach 68 billion USD by 2025 [2]. The aforementioned are only some indicative examples that highlight the enormousness of data currently produced and manipulated. As a matter of fact, according to [3], the volume of data/information created, captured, copied, and consumed worldwide in 2010 was two (2) zettabytes, in 2022 is ninety-seven (97) zettabytes and is expected to reach the number of one hundred and eighty-one (181) zettabytes by the year 2025. This amount of data that is exponentially increased in volume, velocity, and variety, is the so-called Big Data.

Manipulating Big Data in an efficient and effective manner is of great value, because it can enable industries to improve their services by adjusting their decisions that they make based on this data [4]. For that reason, the importance of big data does not revolve around how much data you have, but what you do with it [5]. In general, data can be derived from any source and can be then analyzed to gather insights regarding several aspects such as cost-effective decision making and improvement of provided services. On top of this, especially in the healthcare domain, Big Data can be utilized for performing complex analytics in order to offer personalized healthcare to patients [6] and also lead to the development of effective health policies [7]-[9]. Furthermore, Big Data is utilized in order to improve financial efficiency [10], while it is also exploited to ensure safety in the transportation domain [11]. Nevertheless, it is worth mentioning that there are plenty of challenges that need to be faced when engaging with Big Data. Those challenges mainly refer to aspects like storage, integration, security, and analysis of data, from which Storage is the main challenge that needs to be faced.

The main issue that needs to be addressed is the selection of the most proper database in order to store and then process Big Data. A plethora of databases exist, which serve different purposes and are utilized for different use-cases. The databases thar are most commonly utilized are the relational and the non-relational [12]. Each type has its own advantages and weaknesses, while it also consists of several subtypes, with different specifications. As a result, picking the most proper databases to utilize in a system that generates and/or handles Big Data is not an easy task.

Manuscript received April 26, 2022; revised August 11, 2022.

doi: 10.12720/jait.13.5.423-432

Having said that, in this paper we firstly provide a brief description regarding Big Data and databases in general and corresponding research initiatives. Three (3) of the databases that are utilized on these approaches are then selected and compared with each other by measuring the query time of the CRUD operations, which is widely used for comparing databases [12]. The overall experimentation takes place in a specifically designed working environment that provides the ability of the adaptive allocation and management of infrastructures based on the networking, computing, and storing requirements of each database.

The rest of this paper is organized as follows. Section II firstly provides a brief description regarding Big Data and databases in general and corresponding research initiatives. Three (3) of the databases that are utilized in these approaches are then selected and thoroughly presented in Section III. Section IV describes the procedure followed for conducting the experiments of this comparative study and presents the comparison results. Finally, Section V summarizes the overall work of this paper and provides insights regarding the following steps.

II. Related Work

A. Big Data

The current age is known as the Big Data era since a tremendous amount of data that are complex are exponentially generated [13]. Based on [14], successfully handling Big Data leads to the improvement of provided services and the efficiency of the decisions that are made. The first reason for the enormousness of the Big Data's impact to the daily life is the fact that there exist many devices which are connected to the internet and generate real time data, due to the growth of IoT [15], [16]. Secondly, users are now capable of generating data. Indicative example of such kind of data is data that originates from social networks or website searches [17]. There exist certain characteristics that are highly correlated with Big Data, from which three (3) are thought to be the most important [18]. Big Data includes data of high Volume, Variety and Velocity [19]. The above properties highlight the need for handling Big Data in most proper way, in order to extract knowledge from it and improve provided services.

B. Databases

First of all, a structure that is capable of storing data in an organized manner is called "Database". Databases rely on a set of principles that derive from the CAP theorem [20]. This theorem defines three (3) guarantees, named Consistency, Availability and Partition tolerance [21]. There exist two (2) principles that have been developed based on the aforementioned theorem. The first one ensures the consistency of data, is named ACID and consists of four (4) guarantees titled Atomicity, Consistency, Isolation and Durability. The second one ensures the availability of data and is called BASE [22]. It consists of three (3) guarantees, namely Basic Availability, Soft state, and Eventual consistency [23], [24].

As stated in Section I, databases are split into two (2) categories. Relational databases are part of the first category and rely on the ACID principle [25]. In relational databases data is stored and presented in tables that have rows and columns, as shown in Fig. 1 [26]. Non-relational databases are part of the second category. This one consists of four (4) subcategories [27], as depicted in Fig. 1, and mainly focuses on the BASE principle [28]. The first subcategory is the key/value pair-based, where every value corresponds to a key. The second subcategory is the column-based where every column of a dataset is stored separately [29]. The third one is the graph-based, where data is stored and represented as a graph [30], [31]. Finally, there are the document-based databases where the data are stored in the form of documents that consist of many different key-value pairs [32].

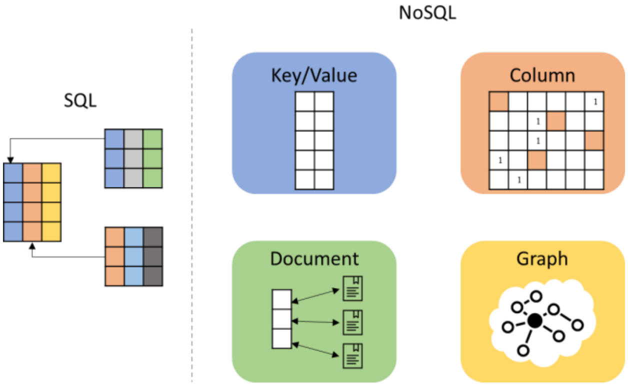


Figure 1. Structure comparison of SQL and NoSQL.

C. Big Data Storage

Finding the most proper database in order to manage Big Data has always puzzled the research community. In [33], the authors made use of MongoDB and MySQL in order achieve handling Big Data in an efficient manner. Moreover, the approach proposed in [34] makes use of MongoDB in order to store data regarding the management of national election data, as it has outstanding performance in comparison with other databases. The authors in [35], conclude that MongoDB is a suitable choice for when handling tremendous amount of unstructured data. In another interesting research [36], the authors make use of MongoDB, on top of which they develop an automated MapReduce framework classifying crime news. Regarding the domain of healthcare, the authors in [37] utilize another NoSQL database named ArangoDB for storing Big Data regarding diabetic children. ArangoDB is also used in [38] storing data from the Italian Business Register, outperforming other databases that were tested during this research. Regarding data that originate from social media, ArangoDB is a great choice when is being used as a graph-based database [39]. CouchDB, which is another document-based database, has also been used for storing Big Data. In [40], the authors integrate this database into a real-time wireless communication system for clinical simulation, as it overcomes latency issues that other tested databases have. The authors in [41] also used the above database for developing a high-performance and flexible chemical structure and data search engine.

Based on the aforementioned approaches, it appears that NoSQL databases are more preferrable for handling Big Data. In deeper detail, the document-based databases are generally preferred over other types [35]. From a plethora of databases that belong to this subtype, we chose MongoDB, ArangoDB and CouchDB because they are open source and well documented. As a result, the performances of those three (3) databases are examined in terms of CRUD operations, by utilizing a dataset from the healthcare domain, in order to conclude which is the most suitable one for handling Big Data. A thorough description of the aforementioned databases is available in Section III.

D. Resources Management

1) Network management

With the ever-increasing network traffic, managing a network faces several new challenges, some of which are highlighted in [42]: complex networks, high costs, slow network construction, slow installation of new services on existing networks, insufficient flexibility in managing network traffic. Due to the aforementioned problems, there is a need for network management, as Software Defined Networks [43]. These types of networks allow applications to directly address such problems by building virtual resources, networks (routers, switches, etc.) but also virtual connections among them and various virtual machines.

The OpenStack tool is available for delivering IaaS (i.e., Infrastructure as a Service) solutions, as a cloud build and management tool. It can manage a large number of resources within the entire data center in which it is located, including processing, storage and network resources. In this context, the authors in [44] have extended the use of the Neutron service (one of the services that builds OpenStack as a whole, which is the networking management service within the virtual cloud that is being built) to the ends of the network that has been built to use this service from IoT devices remote from the existing cloud. In this way, the devices themselves can dynamically manage the cloud through some data-driven decisions. Eventually the above research managed to build a Network as a Service which is hosted within the cloud that has been created and is managed by the devices that exist at its ends (in the Edge Cloud part).

Furthermore, there are three types of networking that occur in digital cloud environments [45]:

- Private clouds: A Cloud Computing environment that exists only for a specific subject (Organization, business, etc.) and is not accessed by external subjects.
- Public clouds: A Cloud Computing environment that can be used by many subjects for the same or different purposes at the same time. It is dangerous to use when data is stored on it, which a subject should not see.
- Hybrid clouds: Combination of Private and Public Clouds. It is a Cloud Computing environment that contains resources that can be used by many subjects, but also resources that exist only for specific subjects from the above.

According to [45], the combination of Private and Public Clouds as Hybrid Clouds, is the solution in terms of data security and speed of processing. Managing data in Private Clouds is a more time-consuming process due to the need to transfer them to and from the cloud in which they are located. The research continues, proposing a data entry solution based on the metadata of their security needs in similar Hybrid Clouds.

2) Storage management

According to [46], Big Data processing is now mostly based on Non-Relational Databases, such as MongoDB, but in addition to these, the use of MySQL is also very common, for greater security and data forecasting. Non-Relational Databases provide ease in scaling the processes that need to be performed to execute the queries they receive. After all, this is main reason why they are often chosen over Relational Databases.

The types of different storage techniques that can be used for different data are as follows [47]:

- Network File System Storage: The placement of data in a hierarchical format, just like the way a computer stores files on its disk, so that the user can easily access this data.
- Block Storage: The placement of data in different blocks, where each block has its own key (identifier). This placement is done in such a way that some pieces of data can be stored even in environments with different operating systems. So instead of having to find an entire path for the stored data, simply referring to the correct identifier is enough to get the data back quickly.
- Object Storage: This storage technique is also the most common for general type (abstract) data (videos, images, etc.). The files given for storage are fragmented and stored within all available hardware. The return of data is done through very specific metadata (metadata) given in the corresponding programming interface (API) (which supports the Object Storage System).

Each different technique has its own advantages but also its own limitations. Although Object Storages cannot update their data, they have low costs in writing and reading their data and are therefore often used by administrators in cloud-centric environments. In the framework "Skedulix" [48] an attempt is made to build a hybrid cloud with the help of Object Storage of MinIO. In this effort, a functional hybrid network was built, as well as the algorithms that run to accelerate workloads from the analysis of the data performed.

It is now advantageous to have a cloud-centric environment in Big Data analytics environments. In [49], an attempt is made to explain a cloud-centric system, for the ideal storage of Big Data, citing and analyzing modern methodologies on the management of this data. Finally, this research addresses several current challenges in the industries:

- Data lifecycle management: Modern Big Data that is generated daily, should be able to each system that manages them to decide whether they will be stored in a private or a public cloud through decisions on the data.
- Data storage: Non-Relational Databases support many advantages for Big Data storage. However, such Databases do not support common SQL queries, and this makes them more difficult than Relational Databases in restoring their data. Automatically selecting and translating non-relational queries into relational queries and vice versa is a modern challenge in the field of Storage Management.
- Data placement: The separation of the data storage part (in which database, in which country, in which continent, in which replica) is a query that can be executed in static ways. However, it is important that this process can be dynamically selected by the respective system that manages the respective data, through criteria that will optimize the needs of this system.

In [50], reference is made to the most real need for more software-defined storage space - hence in virtual computing clouds, explaining that it is now interesting to be able to allocate only the resources required for processing certain data, to maximize the efficiency in the use of all storage and processing resources of a cloud. It focuses on resolving the reliability of big data and reducing the cost of storing it in a cloud database. The proposed technique is predictive and therefore tries to make as few replicas as possible with data that is more likely to be destroyed in the future. The ability to reduce the number and size of replicas on data sets is critical as it saves useful space for the application running this process. Although this research yields an innovative algorithm in the field of Storage Management, no data security techniques are identified.

3) *Computing management*

The management of Big Data processing has now been directed to a more dynamic way of processing it through the computer cloud that manages them. The world belongs to the age of Cloud Computing, due to the ability of cloud computing to easily scale and de-scale the resources they have. In addition, the term Cloud Computing is beneficial to the business sector, as it has the ability through it a business to have on demand storage space, processing power, networks, servers, databases, etc., without owning any of them.

In the model developed in [51], an attempt is made to optimize the production of virtual machines within the built-in cloud, to better process health data from devices connected to the Internet (Internet of Things - IoT). The above effort exploits three optimization algorithms (GA, PSO, PPSO) and manages to surpass existing methodologies by 50% in terms of performing the experiments set in the mentioned model. However, the system that has been designed assumes an infinite number of processing units and is therefore difficult to implement in real environments.

At the same time, the authors in [52] present a set of architectures for creating a private computer network for each user for the purpose of data processing through the OpenStack platform. More specifically, the idea of the architecture is to create a map-graph on which each node is one of the necessary services that must be performed. This set of architectures, which in research are called "Portfolios" and are identified by a code that corresponds to each user, differ in the flow that exists in the execution. That is, a service may lead to another, a service may lead to many others, or a service may require the execution of one or more other services for its successful execution.

Finally, the authors in [53] discuss a new resource management system for Infrastructure-as-a-Service (IaaS) platforms and OpenStack software. More specifically, one of the main problems in managing virtual machines for demanding applications is that in a scenario where many users require multiple resources at the same time, the total execution time of the tasks can be significantly delayed. This is obviously not desirable as it only degrades the user experience. The research presents a resource and virtual machine scheduling system called EJM-CS. In this system idle virtual machines are used to speed up the processes of other tasks in a distributed way and describe in detail the ways to deal with problems such as the different computing power of each virtual machine and the order of selection of systems to create the distributed execution network.

III. Document-Based Databases

MongoDB is a document-based database [26] that stores data as JSON-like documents with dynamic schemas. It focuses on flexibility, speed, power, and ease of use [54]. The indexing of the documents takes place by any field while there is also the possibility to run MongoDB on multiple hosts. It can also support several MapReduce and aggregation tools [55]. However, the retrieval of data from a MongoDB is not an easy task because there is a high probability that queries of excessive complexity should be executed in order to succeed in retrieving the desired documents from the database.

A well-known multi-model database is ArangoDB [56]. It supports both vertical and horizontal scaling, making it suitable for application that need to be scalable [46]. Furthermore, it is less operationally complex because, by being a multi-model database, it allows the users to choose the storage technology that fits the best to the data that they need to manipulate [57]. ArangoDB can be operated as a key/value pair-based, document-based and graph-based database.

CouchDB is also a document-based database. It utilizes the HTTP protocol and the JSON data format, while it also capable of running in a single machine or in a cluster of hosts, depending on the needs of the corresponding application [58], [59]. It also contains some ACID properties as part of its features. A summarization of the main features of the aforementioned databases is presented in Table I.

TABLE I. SUMMARIZATION OF MONGODB, ARANGODB AND COUCHDB FEATURES

Characteristic	ArangoDB	MongoDB	CouchDB
Type	Multi-model	Document-based	Document-based
Query Language	AQL	MQL	JavaScript
Spark	Yes	Yes	Yes
MapReduce	Yes	Yes	Yes
Fault Tolerance	Replication	Replication	Replication
Replication Mode	Master-Slave, Multi-Master	Master-Slave	Multi-Master
Clustering	Yes	Yes	Yes
Language	JavaScript, C++	C++	Erlang
Suggested Use	Combing different data models in one query	Handling scaling data with text, geospatial, time series dimensions	Web use cases and mobile applications

IV. COMPARATIVE STUDY

A. Working Environment

In order to evaluate the performance of the chosen databases, the latter have been parameterized, configured, and finally installed within the context of the Diastema platform [60], offering adjustable network, storage and computing management for the different requirements and needs. Shortly, the overall goal of Diastema is to solve the needs that arise when one manages large volumes of data, providing "Data as a Service". With the primary goal of intelligent and efficient use of resources for the needs of its services, Diastema turns raw data into valuable knowledge. All this is achieved through a series of actions which are formulated and implemented according to the data requirements of each case. In particular, it provides a complete infrastructure management system, which makes use of knowledge extracted from previous applications and infrastructure implementations in the current ones. This complete infrastructure management system is provided as a full stack that facilitates data and application needs.

The above system is a multi-tool that will answer questions:

- Data-centric decisions rather than service-centric solutions. It provides the structure and control of the functions that are responsible for a variety of technical functions. These functions consist of hardware, software, and networking in physical and virtual environments, while their goal is to minimize downtime and maintain productivity. The management of computing resources, storage resources and networking resources is fully efficient and optimized for data functions and data-based applications.
- Perception and use of Data as a Service. Data as a service promotes automation and quality and ensures that the data provided has meaning, value and relevance through approaches to data cleansing, modeling, interoperability, and efficient storage. Data analysis techniques are performed holistically in multiple data stores and locations, along with advanced modeling techniques that define flexible schemas that can be utilized in various processing frameworks.
- Data display. It goes beyond the simple representation of data and their analysis, leading to adaptive visualization in an automatic way according to the analysis of applications and the semantics of data.
- Data analysis and their analysis techniques. It allows openness and scalability by providing an environment for data scientists and professionals to easily implement their data analysis functions using an illustrative example, as well as to define their preferences and limitations. At the same time, advice is provided to the infrastructure management system on how best to carry out these analyzes.
- Flexible process modeling to execute. It allows flexible modeling of processes based on functionality, which are mapped in an automated way. The results of the analysis provide feedback to business analysts with specific suggestions for optimizing and customizing the overall process.
- Data-centric application sizing. It facilitates the analysis of data-focused applications in terms of predicting the required data services, their interdependencies with the application micro-services, and the required key resources. Thus, by allowing the identification of application data-related properties and their data needs, it is able to provide specific performance and quality guarantees.

Fig. 2 depicts a high-level architecture of the Diastema platform, including the different layers of the Infrastructure, the API, and the User Interaction.

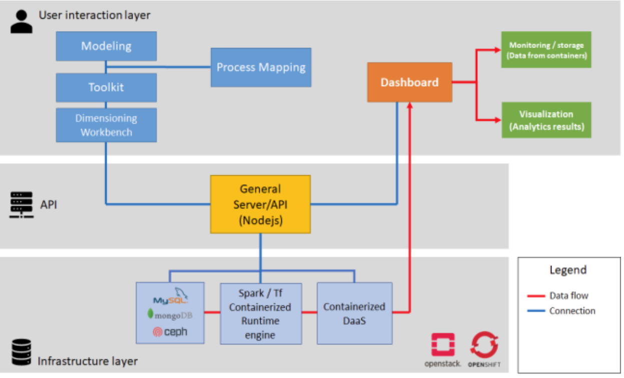


Figure 2. Diastema high-level architecture.

The project environment is depicted in three layers of technologies, where each layer serves a specific purpose (Fig. 2), as they were analyzed at the initial level:

- Beginning from the bottom, the first layer (Infrastructure layer) is the foundation of the project as all the services of the application are based on it and are developed. It includes containers with databases as well as the corresponding technologies needed for data processing. It also includes tools for machine learning as well as functions such as modeling, cleaning, and data analysis.

- In the middle layer (Programming Interface Layer - API) is the central API that is responsible for the interconnection of all services. It is the "brain" of the application, as it has the role of the central flow management unit, performing the corresponding action at any time.
- Finally, in the higher layer (User Interaction layer) there is the point of contact with the user and her data needs. At the beginning there exists the modeling and at the same time the multi-tool (Toolkit) which are the initial interface that the user encounters, defining her needs and possible limitations.

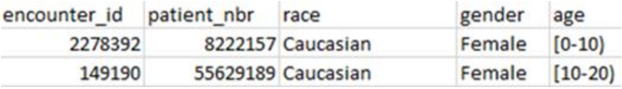
Modeling mainly concerns the user with the role of Business Analyst, who gives an overview of the needs and requirements of her data. This information continues at the Toolkit level. The Toolkit concerns the user with the role of Data Scientist. It takes the information provided by the Business Analyst and builds on it, the technical data requirements. It is able to set limits on both the time and the efficiency of the whole process. Between these two interfaces, there is process mapping, which uses automated machine learning to automate the needs of the Business Analyst and the requirements needed, if the user does not have the necessary know-how or time for extensive research to select the most appropriate algorithm. The Dimensioning Workbench is used in the dimensioning of services to predict the required resources, so that they can then be sent to the infrastructure management service. Finally, in the Dashboard the user can see the result of the whole process, using data visualization technologies to better understand them. This is an Environment for the display of data and resources needed for each process. It displays the result of the data processing as it simultaneously receives data from the containers management service.

B. *Databases Setup*

Based on the aforementioned, the three databases (i.e., MongoDB, ArangoDB and CouchDB) have been installed within a virtual machine deployed in the infrastructure layer, in order for the further analysis of different CRUD operations along with the other involved layers of the Diastema platform. The virtual machine's operating system was Ubuntu 20.04.2.0 LTS, while the RAM size was altered during the experimentations. More specifically, the RAM sizes were 8GB, 16GB and 32GB, thus examining the databases' requirements for resources.

C. *Overall Experiment*

The evaluation of the databases' performance was carried out by performing CRUD operations of different complexity and measuring the average corresponding response time. In order to perform the aforementioned task, the Python programming language and the corresponding databases' drivers were utilized. The above approach was selected in order simulate a real use-case scenario, were a third-party application makes use of a database. It is worth mentioning that each database provides its own tools in order to evaluate its performance. However, the results would not be as objective as possible, given that these tools were utilized for the evaluation of the database's performance. Furthermore, in each database the same dataset was stored. The size of this dataset is almost one (1) TB and consists of one hundred and seventeen (117) unique features and roughly seventy-four million (74.000.000) records. Each record represents a specific encounter (i.e., visit to hospital), while the dataset originates from the Health Facts database [61] that consists of electronic medical records and includes information such as encounter data, provider specialty, demographics, diagnoses laboratory data and pharmacy data. An instance of this data is depicted in Fig. 3, where a subset of the features and the records is available. For example, the first record shown in Fig. 3 refers to a hospital visit (i.e., encounter) of a patient with a unique identifier (i.e., "patient_nbr"), who is Caucasian, female and her age is between ten (10) and twenty (20) years old.



encounter_id	patient_nbr	race	gender	age
2278392	8222157	Caucasian	Female	[0-10)
149190	55629189	Caucasian	Female	[10-20)

Figure 3. Dataset example.

1) *Operation of Create*

As for the first implemented operation (i.e., Create), the execution times for a corresponding query in the chosen databases and for different amounts of RAM are shown in Fig. 4, while a query instance for every database using Python, in order to add a patient into the database, is shown in Table II. It appears that, in this case, CouchDB is not able to compete with the other two (2) databases.

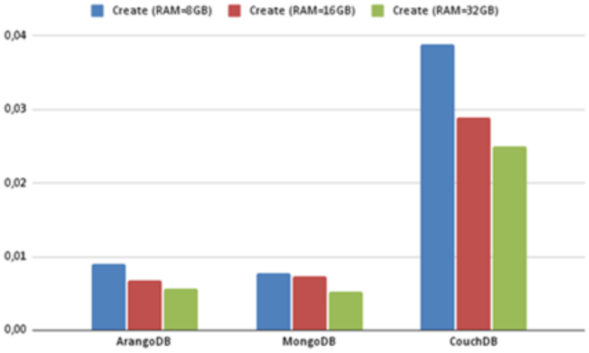


Figure 4. Average execution time in seconds of Create operations.

TABLE II. INDICATIVE CREATE QUERIES

Database	Query
ArangoDB	doc1["id "] = "12345", doc1.save()
MongoDB	data.insert_one({ "id": "12345", "race": "Caucasian"})
CouchDB	doc_id, doc_rev = couch_db.save(doc1)

2) Operation of Read

As for the second implemented operation (i.e., Read), the execution times for a corresponding query in the chosen databases and for different amounts of RAM are shown in Fig. 5. Based on the results, MongoDB performs the best in Read operations. An example of a read (i.e., select) query in every database using Python, for retrieving patients based on their unique identifier, is shown in Table III.

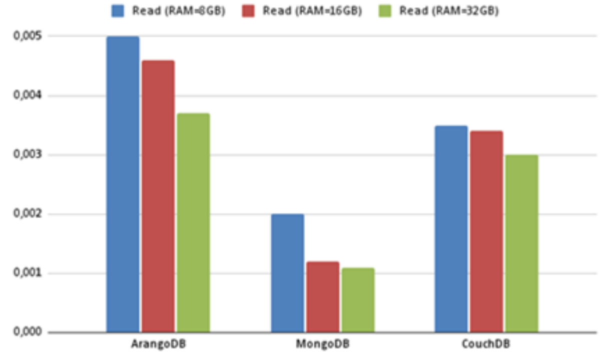


Figure 5. Average execution time in seconds of Read operations.

TABLE III. INDICATIVE READ QUERIES

Database	Query
ArangoDB	test_patient = diabetic_collection['2790']
MongoDB	patient = data.find_one({'encounter_id': '2790'})
CouchDB	db.get('2790')

3) Operation of Update

As for the third implemented operation (i.e., Update), the execution times for a corresponding query in the chosen databases and for different amounts of RAM are shown in Fig. 6, whilst an example of those queries, for updating the ‘race’ feature of a patient, is shown in Table IV. According to the below graph, ArangoDB’s performance is outstanding when it comes to updating a record in the database.

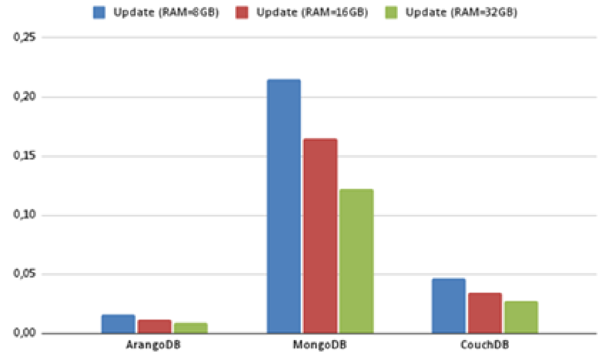


Figure 6. Average execution time in seconds of Update operations.

TABLE IV. INDICATIVE UPDATE QUERIES

Database	Query
ArangoDB	diabetic_collection['test_subject'] ['race'] = 'Caucasian'
MongoDB	raw_data.update_one({ "encount_id": "12345" }, { "$set": { "race": "Caucasian" } })
CouchDB	self.db.get(12345) ["race"] = json["Caucasian "]

4) Operation of Delete

Concerning the fourth implemented operation (i.e., Delete), the execution times for a corresponding query in the chosen databases and for different sizes of RAM are depicted in Fig. 7. In that case, ArangoDB outperforms the other databases.

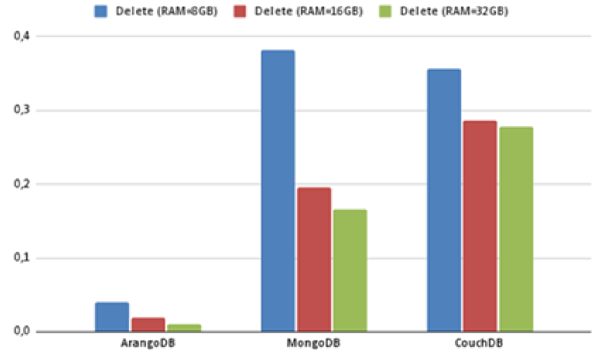


Figure 7. Average execution time in seconds of Delete operations.

TABLE V. INDICATIVE DELETE QUERIES

Database	Query
ArangoDB	diabetic_collection["12345"].delete()
MongoDB	aw_data.delete_one({"encounter_id": "12345"})
CouchDB	db.delete(dict(type=patient, encounter_id=’12345’))

D. Discussion of Comparative Study

The findings of this study suggest that, as for the Create and Read operations, MongoDB achieved the best performance. This means that, in a corresponding application where the insertion of data is a priority, MongoDB should be chosen. Regarding the Update and Delete operations, ArangoDB outperformed the other databases, meaning that in a corresponding application where altering the stored data is a must, ArangoDB should be utilized. As for CouchDB, its performance can be explained by the fact that it depends on the HTTP, that is thought to be a high latency protocol. (Table V)

V. CONCLUSION

In this day and age there is a colossal amount of data that is generated and/or processed in a daily basis. This amount of data has led to the introduction of the term of Big Data. Data is valuable because, if manipulated correctly, knowledge can be extracted from it and as a result, provide useful insight and improve decision making. However, before reaching the point of extracting useful knowledge from data, a number of steps should be followed. Among these steps, the is one called Data Storage that refers to the process of selecting the most proper database in order to store the data in the most efficient manner possible. This paper presented the main principles of Big Data and Data Storage in general, focusing on the databases that are currently in use in order to address the problem. Three (3) of those were chosen and compared in this study. The results showed that MongoDB achieved the best performance in the Create and Read operations while ArangoDB outperformed the other databases in the Update and Delete operations.

Based on all the above, we plan to expand our research by testing those databases in real time situations, where streaming data are stored and processed. We also hope to add more databases in our trials, especially those that have recently been introduced and, as a result, have not been thoroughly examined by the research community.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Konstantinos Mavrogiorgos conducted the research; Athanasios Kiourtis and Argyro Mavrogiorgou analyzed the data; Argyro Mavrogiorgou conducted the experiments and the evaluation study; Konstantinos Mavrogiorgos, Athanasios Kiourtis and Argyro Mavrogiorgou wrote the paper; Athanasios Kiourtis proofread the manuscript; Dimosthenis Kyriazis supervised the overall research; All authors had approved the final version.

Acknowledgment

The research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code: DIASTEMA - T2EDK-04612).

References

[1] I. Ajah, *et al*., "Big data and business analytics: Trends, platforms, success factors and applications," *Big Data and Cognitive Computing*, vol. 3, no. 2, p. 32, 2019.

[2] Statista - Big data - Statistics & Facts. [Online]. Available: https://www.statista.com/topics/1464/big-data/#dossierKeyfigures

[3] Statista - Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025. [Online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created/

[4] S. Shamim, *et al*., "Role of big data management in enhancing big data decision-making capability and quality among Chinese firms: A dynamic capabilities view," *Information & Management*, vol. 56, no. 6, p. 103135, 2019.

[5] E. Kapassa, *et al*., "An innovative ehealth system powered by 5G network slicing," in *Proc. Sixth Int. Conf. on Internet of Things: Systems, Management and Security*, 2019. p. 7-12.

[6] V. Jagadeeswari, *et al*., "A study on medical internet of things and big data in personalized healthcare system," *Health Information Science and Systems*, vol. 6, no. 1, pp. 1-20, 2018.

[7] S. Kbioassist, *et al*., "Crowdhealth: Holistic health records and big data analytics for health policy making and personalized health," *Informatics Empowers Healthcare Transformation*, vol. 238, p. 19, 2017.

[8] D. Kyriazis, *et al*., "The CrowdHEALTH project and the hollistic health records: Collective wisdom driving public health policies," *Acta Informatica Medica*, vol. 27, no. 5, p. 369, 2019.

[9] D. Kyriazis, *et al*., "PolicyCLOUD: Analytics as a service facilitating efficient data-driven public policy management," in *Proc. IFIP Int. Conf. on Artificial Intelligence Applications and Innovations*, 2020. p. 141-150.

[10] O. Masters, *et al*., "Towards a homomorphic machine learning big data pipeline for the financial services sector," *Cryptology ePrint Archive*, 2019.

[11] Y. Lian, *et al*., "Review on big data applications in safety research of intelligent transportation systems and connected/automated vehicles," *Accident Analysis & Prevention*, vol. 146, p. 105711, 2020.

[12] S. Bjeladinovic, "A fresh approach for hybrid SQL/NoSQL database design based on data structuredness," *Enterprise Information Systems*, vol. 12, no. 8-9, pp. 1202-1220, 2018.

[13] N. Deepa, *et al*., "A survey on blockchain for big data: Approaches, opportunities, and future directions," *Future Generation Computer Systems*, 2022.

[14] Gartner Glossary – Big Data. [Online]. Available: https://www.gartner.com/en/information-technology/glossary/big-data

[15] M. Marjani, *et al*., "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247-5261, 2017.

[16] R. Habeeb, *et al*., "Real-Time big data processing for anomaly detection: A survey," *International Journal of Information Management*, vol. 45, pp. 289-307, 2019.

[17] S. Abkenar *et al*., "Big data analytics meets social media: A systematic review of techniques, open issues, and future directions," *Telematics and Informatics*, vol. 57, p. 101517, 2021.

[18] Z. Sun, *et al*., "Big data with ten big characteristics," in *Proc. the 2nd International Conference on Big Data Research*, 2018, pp. 56-61.

[19] Z. Sun, *et al*., "Big data with ten big characteristics," in *Proc. the 2nd Int. Conf. on Big Data Research*, 2018. pp. 56-61.

[20] E. Lee, *et al*., "Quantifying and generalizing the CAP theorem," arXiv preprint arXiv:2109.07771, 2021

[21] S. Gilbert, *et al*., "Perspectives on the CAP theorem," *Computer*, vol. 45, no. 2, pp. 30-36, 2012.

[22] A. Meier, *et al*., *SQL & NoSQL Databases*, Springer Fachmedien Wiesbaden, 2019.

[23] M. Mus, "Comparison between SQL and NoSQL databases and their relationship with big data analytics," 2019.

[24] D. G. Chandra, "BASE analysis of NoSQL database," *Future Generation Computer Systems*, vol. 52, pp. 13-21, 2015.

[25] E. Kopic, *et al*., "In-memory databases and their impact on our (future) organizations," In *The Impact of Digital Transformation and FinTech on the Finance Professional*, Palgrave Macmillan, Cham, 2019, pp. 357-370.

[26] C. Győrödi, *et al*., "A comparative study: MongoDB vs. MySQL," in *Proc. 13th Int. Conf. on Engineering of Modern Electric Systems*, 2015, pp. 1-6.

[27] M. Diogo, *et al*., "Consistency models of NoSQL databases," *Future Internet*, vol. 11, no. 2, p. 43, 2019.

[28] V. Sharma, *et al*., "Sql and nosql databases," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 8, 2012.

[29] A. H. Abed, "Big data with column oriented NOSQL database to overcome the drawbacks of relational databases," *Int. J. Advanced Networking and Applications*, vol. 11, no. 5, pp. 4423-4428, 2020.

[30] N. Patil, *et al*., "A survey on graph database management techniques for huge unstructured data," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 2, p. 1140, 2018.

[31] R. Angles, "The property graph database model," in *Proc. AMW*, 2018.

[32] B. Bialek, "MongoDB: The journey from a relational to a document-based database for FIS balance sheet management," in *The Impact of Digital Transformation and Fintech on the Finance Professional*, Palgrave Macmillan, Cham, 2019, pp. 371-380.

[33] B. E. James, *et al*., "Hybrid database system for big data storage and management," *International Journal of Computer Science, Engineering and Applications*, vol. 7, 2017.

[34] L. G. Wiseso, *et al*., "Performance analysis of Neo4j, MongoDB, and PostgreSQL on 2019 national election big data management database," in *Proc. 6th Int. Conf. on Science in Information Technology*, 2020, pp. 91-96.

[35] K. Anusha, *et al*., "Comparative study of MongoDB vs Cassandra in big data analytics," in *Proc. 5th Int. Conf. on Computing Methodologies and Communication*, 2021, pp. 1831-1835.

[36] K. Santhiya, *et al*., "An automated MapReduce framework for crime classification of news articles using MongoDB," *International Journal of Applied Engineering Research*, vol. 13, no. 1, pp. 131-136, 2018.

[37] A. Bazila *et al*., "Prediction of children diabetes by autoregressive integrated moving averages model using big data and not only SQL," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 8, pp. 3510-3513, 2019.

[38] N. Ferro, *et al*., "Graph databases benchmarking on the Italian business register," in *Proc. SEBD*, 2018.

[39] N. S. Patil, *et al*., "A survey on graph database management techniques for huge unstructured data," *International Journal of Electrical & Computer Engineering*, 2018, pp. 2088-8708.

[40] Y. Alhomsi, *et al*., "CouchDB based real-time wireless communication system for clinical simulation," in *Proc. IEEE 20th Int. Conf. on High Performance Computing and Communications; IEEE 16th Int. Conf. on Smart City; IEEE 4th Int. Conf. on Data Science and Systems*, 2018, pp. 1094-1098.

[41] R. Z. Li, *et al*., "A high-performance and flexible chemical structure & data search engine built on CouchDB & ElasticSearch," *Chinese Journal of Chemical Physics*, vol. 31, no. 3, p. 341, 2018.

[42] L. Xiang, *et al*., "Hybrid cloud networking design based on Openstack architecture," in *Journal of Physics: Conference Series*, IOP Publishing, 2020. p. 012011.

[43] C. Monsanto, *et al*., "Composing software defined networks," in *Proc. 10th Symposium on Networked Systems Design and Implementation*, 2013, p. 1-13.

[44] Z. Benomar, *et al*., "Extending openstack for cloud-based networking at the edge," in *Proc. IEEE Int. Conf. on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data*, 2018, p. 162-169.

[45] X. Xu, *et al*., "Data placement for privacy-aware applications over big data in hybrid clouds," *Security and Communication Networks*, 2017.

[46] R. Deari, *et al*., "Analysis and comparison of document-based databases with SQL relational databases: MongoDB vs MySQL," in *Proc. the Int. Conf. on Information Technologies*, 2018, p. 1-10.

[47] Amazon Web Services – "What Is Cloud Storage? | AWS". (2021). [Online]. Available: https://aws.amazon.com/what-is-cloud-storage/

[48] A. Das, *et al*., "Skedulix: Hybrid cloud scheduling for cost-efficient execution of serverless applications," in *Proc. IEEE 13th Int. Conf. on Cloud Computing*, 2020, pp. 609-618.

[49] S. Mazumdar, *et al*., "A survey on data storage and placement methodologies for cloud-big data ecosystem," *Journal of Big Data*, vol. 6, no. 1, pp. 1-37, 2019.

[50] S. Shrivastana, *et al*., "Efficient storage management framework for software defined cloud," *International Journal of Internet Technology and Secured Transactions*, vol. 7, no. 4, pp. 317-329, 2017.

[51] M. Elhoseny, *et al*., "A hybrid model of internet of things and cloud computing to manage big data in health services applications," *Future Generation Computer Systems*, vol. 86, pp. 1383-1394, 2018.

[52] H. Yang, *et al*., "Research on multiple complex data processing methods based on OpenStack cloud platform," *International Journal of Advanced Engineering Research and Science*, vol. 4, no. 4, p. 237113, 2017.

[53] S. Han, *et al*., "An efficient job management of computing service using integrated idle VM resources for high-performance computing based on OpenStack," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4388-4407, 2019.

[54] A. Boicea, *et al*., "MongoDB vs Oracle--database comparison," in *Proc. Third Int. Conf. on Emerging Intelligent Data and Web Technologies*, 2012, pp. 330-335.

[55] MongoDB – Map Reduce. [Online]. Available: https://docs.mongodb.com/manual/core/map-reduce/

[56] Lu, J., *et al*., "Multi-model databases: A new journey to handle the variety of data," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1-38, 2019.

[57] Mindk - ArangoDB: A perfect database for projects with a high level of uncertainty. [Online]. Available: https://www.mindk.com/blog/arangodb/

[58] R. Kumbhare, *et al*., "Tamper detection in MongoDB and CouchDB database," in *Proc. International Conference on Computational Science and Applications*, 2020, pp. 109-117.

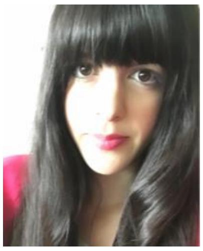
[59] HG Insights – Companies Currently Using CouchDB. [Online]. Available: https://discovery.hgdata.com/product/couchdb

[60] Diastema Platform. [Online]. Available: https://diastema.gr/

[61] B. Strack, *et al*., "Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records," *BioMed Research International*, 2014.

Konstantinos Mavrogiorgos is a postgraduate student in the Department of Digital Systems of the University of Piraeus. His general research interests are based on the fields of data management and analysis. In this context, he participates in the National project beHEALTHIER, conducting research to address issues related to the integration and analysis of data, in conjunction with their purification and reliability. Moreover, he participates in the European project InteropEHRate, conducting research regarding the implementation of D2D (Device to Device) and RDS (Research Data Sharing) protocols that are being developed in the scope of this project.

Athanasios Kiourtis is a Postdoctoral Researcher at the University of Piraeus, Greece (Department of Digital Systems) with a focus on Digital Transformation and Optimus Data Exchange protocols. He received his BSc in Digital Systems in 2013 and MSc in Network-Oriented Systems in 2015, both from the Department of Digital Systems at the University of Piraeus. Since 2019, he holds a PhD from the University of Piraeus (Department of Digital Systems) in the field of Interoperability, with a focus on data transformation and interoperability services applied to heterogeneous healthcare infrastructures and applications. His general research interests lie with issues related to digital transformation, data exchange protocols, data and services interoperability and heterogeneity, big data analytics, as well as cloud services and infrastructures. In this context, he has participated in several EU and National funded projects (e.g. PolicyCLOUD, InteropEHRate, CrowdHEALTH), leading research for addressing issues related to data interoperability and analysis, as well as health information exchange, while he has also contributed to the aforementioned fields through multiple Int. Conf. and International Journal publications. Regarding his Academic expertise, he has collaborated as an Academic Associate with the University of West Attica, Greece (Department of Industrial Design and Production Engineering, Department of Informatics and Computer Engineering), the National and Kapodistrian University of Athens, Greece (Faculty of Nursing), the National Technical University of Athens, Greece (School of Electrical and Computer Engineering), and the University of Piraeus, Greece (Department of Digital Systems).



Argyro Mavrogiorgou is a Postdoctoral Researcher at the University of Piraeus, Greece (Department of Digital Systems). She received her BSc in Digital Systems in 2013 and MSc in Network-Oriented Systems in 2015, both from the Department of Digital Systems at the University of Piraeus. Since 2019, she holds a PhD from the University of Piraeus (Department of Digital Systems) in the fields of Devices and Data Integration, focusing on Internet of Things heterogeneous devices' accessibility and management towards the successful data ingestion and integration. Her general research interests rely on the fields of data management across the complete data lifecycle, and hybrid infrastructures including cloud/edge computing environments, Internet of Things, and cyber-physical systems. In this context, she has participated in several relevant EU and National funded projects (e.g. PolicyCLOUD, InteropEHRate, CrowdHEALTH), leading research for addressing issues related to data integration and analysis, as well as data cleaning and reliability, while she has also contributed in the aforementioned fields through multiple Int. Conf. and International Journal publications. Regarding her Academic expertise, she has collaborated as an Academic Associate with the University of West Attica, Greece (Department of Industrial Design and Production Engineering, and Department of Electrical and Electronics Engineering), the National and Kapodistrian University of Athens, Greece (Faculty of Nursing), the National Technical University of Athens, Greece (School of Electrical and Computer Engineering), and the University of Piraeus, Greece (Department of Digital Systems).

Dimosthenis Kyriazis is an Associate Professor at the Department of Digital Systems of University of Piraeus. He received his BSc from the Department of Electrical and Computer Engineering of the National Technical University of Athens in 2001 and his MSc in Techno-economics in 2004. He holds a PhD from the school of Electrical and Computer Engineering Department of NTUA in the fields of Service Oriented Architectures, focusing on quality aspects and workflow management since 2007. His general research interests rely on service-oriented, distributed and heterogeneous systems, software engineering and data management. In this context, he has taken part in several EU and National funded projects such as BigDataStack, CrowdHEALTH, MATILDA where he has led research regarding aspects that relate to quality-of-service provisioning, fault tolerance, workflow management, performance modeling in service oriented environments. Other application domains that his research focuses on are multimedia, post-production, virtual reality, finance and e-health. Currently, he is focusing on data management, virtualization technologies for high-availability in cloud and edge environments, as well as socially enhanced techniques for IoT management. In this context, he coordinates EU funded projects, while he also analyzes topics related to Big Data management and content syndication. Finally, he has participated is several EU working groups such as Future Internet Architecture and Cloud QoS&SLAs.