COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 132 (2016) 75–82

ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

An effective model for store and retrieve big health data in cloud computing

ARTICLE INFO

Article history:
Received 4 September 2015
Received in revised form
8 April 2016
Accepted 11 April 2016

Keywords:
Health data
Relational database
NoSQL
Big data
Information storage and retrieval
Cloud computing

ABSTRACT

Background and objective: The volume of healthcare data including different and variable text types, sounds, and images is increasing day to day. Therefore, the storage and processing of these data is a necessary and challenging issue. Generally, relational databases are used for storing health data which are not able to handle the massive and diverse nature of them.
Methods: This study aimed at presenting the model based on NoSQL databases for the storage of healthcare data. Despite different types of NoSQL databases, document-based DBs were selected by a survey on the nature of health data. The presented model was implemented in the Cloud environment for accessing to the distribution properties. Then, the data were distributed on the database by applying the Shard property.
Results: The efficiency of the model was evaluated in comparison with the previous data model, Relational Database, considering query time, data preparation, flexibility, and extensibility parameters. The results showed that the presented model approximately performed the same as SQL Server for "read" query while it acted more efficiently than SQL Server for "write" query. Also, the performance of the presented model was better than SQL Server in the case of flexibility, data preparation and extensibility.
Conclusions: Based on these observations, the proposed model was more effective than Relational Databases for handling health data.

1. Introduction

By development of technology, the request for storing and processing large amounts of data is increasing excessively. The volume of data is enormous right now; and it is predicted to reach 35 zettabytes by 2020 [1]. Huge and increasing amount of data are seen in all branches of science, specifically in healthcare. The volume of worldwide healthcare data was equal to 500 petabytes in 2012; and it is expected to be 25,000 petabytes in 2020. In addition, there are various types of data in healthcare including personal medical records, radiology images, clinical trial data, FDA submissions, human genetics and population data, 3D imaging, sensor readings, genomics, etc. [2,3].

These complex data are collected from different sources with several structures. It is a difficult task to store and analyze medical data. However, cumulative analysis of healthcare is necessary for discovering new useful patterns, recognizing unknown relationships, making universal decisions, identifying effective treatments and selecting best practices for a group of patients. Therefore, an effective method is needed for health data management in order to fulfill the requirements in this branch. Moreover, the model should be available and consistent in voluminous data [4,5].

In recent years, maintenance and processing of various and high volume data have created the "Big Data" challenge. As Gartner said: *"Big Data is high-volume, -velocity and -variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making"* [6].

Generally, most organizations store their data in relational database management systems (RDBMS) [7]. There are some restrictions on using relational databases (DBs) for Big Data including efficient processing, effective parallelization, scalability and costs [7–9]. For example, users must convert Big Data into tables with pre-design fields; and it creates a complex and difficult structure of database that works slowly. In addition, data cannot fit into a table simply [7]. As another restriction, attaching tables to the distributed system is not easy. Therefore, these databases do not easily act in a distributed system [7,10]. On the other hand, due to scalability and availability requirements, it is inefficient to store massive amount of data locally. Another restriction is inability of RDBMS for storing huge data due to its limitation on database size. Moreover, companies must focus on providing a lot of management resources to store massive data rather than focusing on their business innovation; and it consumes a lot of time and cost.

Recently, some new technologies such as a new class of databases, known as NoSQL, and a computing model, called Cloud Computing, have been developed to resolve these problems. NoSQL databases provide better performance for storing and maintaining large scale data regardless of their formats. These databases can also operate without predefined schema and relationship [5,7,11–13]. It provides high throughput for voluminous and heterogeneous data in a distributed environment [5]. Moreover, users do not need to be familiar with SQL language.

Due to their simple data models, NoSQL databases are used easily. These databases can function in a distributed manner. So, users can scale a single database on additional inexpensive machines instead of a more powerful and costly single machine. Data processing in NoSQL databases is generally faster than relational databases. Unstructured, semi structured data and data with various form and size are effectively used in these databases [7,14].

Another new technology mentioned above is Cloud Computing. This computing model is a new paradigm that reduces the costs for management of resources such as hardware and software by offering resources on the network. So, users may access to demand services everywhere in the world. Cloud is a kind of parallel and distributed systems with a collection of computers. It provides computing resources based on service-level agreements established between the service provider and users. Cloud environments have some advantages such as availability, scalability, performance, multi-tenancy, elasticity, fault tolerance and load balancing. This computing model enables users to consume computing resources (e.g., networks, servers, storage, applications and services) as a utility over the Internet. Instead of local servers or personal devices, resources are shared with minimal management effort to handle applications [11,15].

In this paper, based on NoSQL technology, an efficient model has been proposed by considering the essential requirements of health data. The model is compatible with Cloud environment and utilizes capabilities of Cloud. In the following, its performance has been compared with former model considering query time in some workload, data preparation, flexibility and extensibility.

The rest of the paper is ordered as follows. Section 2 offers an overview on different approaches to handle and maintain big data such as health data. Section 3 describes the nature of health data and presents appropriate model for it. The performances of former and new model are discussed in section 4. Subsequently, a conclusion is given in section 5.

2. Background

In the big data revolution, health sector is a notable aspect. The volume of health data is huge and increases explosively. On the other hand, these data are complex in the nature and very difficult to analyze. Medical data are generated from multiple sources in different forms. Healthcare organizations must convert the information into useful knowledge through operational studies, research, and tools. Comprehensive analysis of healthcare-related data provides desirable results that can improve the quality of decision making process. For example, side effects of some drugs are indicated via analyzing patient histories [5,16].

Most medical storage systems are based on relational databases, which are not efficient and sufficiently flexible, according to the nature of these data [5]. Recently, the use of Cloud Computing has become epidemic in healthcare. Wang et al. [17] introduced some important Big Data applications in healthcare domain including large datasets for health information systems (HIS) and clinical decision support systems (CDSS), Medical Body Area Networks (MBANs). They offered Cloud Computing infrastructures for designing and developing Big Data Analytics due to its conjunction with fast communication networks, programming models, semantic web, and machine learning algorithms [17]. In a study, Xbase, a hybrid approach based on RDBMS and Hadoop was proposed, which resulted in higher performance and lower costs. The system was implemented in a Cloud Computing infrastructure that provided infinite distributed storage and computation capability [18].

Microsoft Health Vault, Google Health, Dossia, and Mphrx are some public health management systems based on Cloud Computing. These systems are capable to store and maintain personal data of patients through an online access. So, users can retrieve their data anytime and anywhere from every device. Also, they can take care of themselves via these systems according to their health status. These cloud services are a kind of Software as a Service (SaaS) platform. Despite, Google Health has been permanently stopped from 2013, while Microsoft Health Vault, Dossia, and Mphrx are still available [19–22].

In order to process and analyze large genomic datasets, Cloud Computing and Hadoop have been introduced in the study done by O'Driscoll as proper technologies. Genome data are significant in healthcare branch that can add much more value in this sector. This model provides a distributed and parallelized infrastructure for data sets with petabyte scale [23]. Nguyen et al. proposed an approach based on the Apache HBase (a type of NoSQL DBs) and the Map-Reduce programming to store and process clinical data. This system is integrated with a web-based layer to parallelize the computation process [24].

Doukas et al. [25] presented the implementation of a mobile system based on Cloud Computing that enables electronic healthcare data storage, update and retrieval. Amazon's S3 cloud service was established in this project which provided online management of patient's data. In other study [26], the authors addressed a gap between potential and actual data usage by focusing on open, visual environments. A framework was developed for efficient use of healthcare data by integrating the MIMIC database in a RapidMiner environment. Also, Hadoop and some analytic algorithms were used for data analysis.

In a study [27], the architecture of healthcare SaaS Platform was analyzed for Decision Support Service. Microsoft's Azure was introduced as a Cloud service in this model. Vukićević et al. [28] propose a cloud-based system for the analysis of biomedical data. This system integrated meta-learning framework in order to select the best predictive algorithms and open source big data technologies for analysis. In another research, a communicational framework was proposed which related key segments of health world. The data could be entered in this framework and mine by data mining algorithms [29]. Above studies are classified in Table 1 clearly.

In the above healthcare studies, some restrictions exist for Big Data. As previously noted, healthcare data have a high and rising volume extracted from different sources. Therefore, it is necessary to choose a model with distribution capability for them. The implicit RDBMS models don't fulfill these requirements. Also, most proposed approaches are limited to SaaS and cannot be used for backend users. The other aspect that has to be considered in healthcare data models is various types of data. Therefore, we present a comprehensive model in order to overcome these limitations on healthcare's applications.

3. Method

3.1. NoSQL database modeling

In the last few years, usage of NoSQL database has been spread in different sectors because of their ability to deal with new requirements of applications. Also, NoSQL databases are becoming the preferred selection for storing significant amounts of redundant data in the Cloud. NoSQL DBs present new storage architectures that provide high scalability, availability and fast retrieval requirements for managing unstructured and partially structured data. Many NoSQL DBs are open source and they are cheaper per terabyte than traditional DBs. These DBs are more appropriate for the web based data [14].

Based on the data model, there are four general types for these databases: Key-value DB, Columnar DB, Document-Based DB and Graph-Based DB [8]. Key-value DBs have the least complex structure. In this model, data are stored and retrieved by predefined key with an independent value. The key of a key/value pair is a unique value in a set and it is used for accessing to the data [14]. In Columnar DBs, the column is the major part. It consists of related data grouped closely. The data are stored in a column-family basis that is typically defined at the configuration or startup time. Columns can store any data types effectively [8]. Document-based DBs potentially organize huge and complex documents. This model supports multiple types of massive documents with flexibility to add any numbers of fields in any length. The document is considered as a whole object; and it isn't split into name/value pairs. Document databases allow indexing of documents on both primary identifier and its properties [7,8,14]. Graph-based databases use graph structures with nodes, edges and properties for storing data [30].

Hence, there is a long variety of NoSQL databases in different data models. Users must select one of them based on their application requirements. In the next section, we are going to argue about an appropriate NoSQL database for health data, depending on data characteristics.

3.2. Appropriate model for health data

As mentioned in the previous sections, relational DBs are not proper choices for handling health data, assuming their special nature. This chapter describes an appropriate method for handling these data considering new database models.

Table 1 – The studies on Cloud Healthcare Systems.

Cloud Services/Models	Year	Type of services	Purpose	Implementation	Tools	Type of data	Usage
Dossia [20]	2006	SaaS	An online individual healthcare system	Available	Not declared	PHR[a]	Personal
Microsoft Health Vault [21]	2007	SaaS	An online individual healthcare system	Available	Not declared	PHR	Personal
Google Health [22]	2008	SaaS	An online individual healthcare system	Stopped From 2013	Not declared	PHR	Personal
Xbase [18]	2010	A model	Storage and Analysis of healthcare data	No	RDBMS and Hadoop	MHR[b]	General
Doukas [25]	2010	SaaS	Storage, update and retrieval of EHR	Yes	Android OS and Amazon's S3	PHR	Personal
Mphrx [19]	2011	SaaS	An online individual healthcare system	Available	Not declared	PHR	Personal
Nguyen [24]	2011	A model	Storage and process of clinical signal data	No	Hbase and Map-Reduce	Clinical signal data	General
O'Driscoll [23]	2013	A model	Introducing some tools to analyze large genomic datasets	No	Hadoop	Genome data	General
Vukićević [28]	2014	A model	Analysis of biomedical data	No	Some algorithms	Biomedical data	General
Oh [27]	2015	SaaS	Offer a communication model for health Cloud	No	Microsoft's Azure	EHR[c]	General
Parekh [29]	2015	A model	Offer a communication model for mobile health	No	Open Stack and Java language	Data from Healthcare systems	General
Poucke [26]	2016	A model	Introducing a model to analyze health data	Yes	Hadoop and RapidMiner	MIMIC data[d]	General

[a] Personal Health Record.
[b] Medical Health Record.
[c] Electronic Health Record.
[d] A database of physiologic data.

Table 2 – The comparison of healthcare data and NoSQL DBs characteristics.

Healthcare data characteristics	NoSQl DBs characteristics			
	Key-value DBs	Document-based DBs	Column-based DBs	Graph-based DBs
Mostly document based	Storing key and its dependent value	Storing of documents	Storing key and its dependent value	Storing nodes and their relationships
Different types of data	Storing different types	Storing different types	Storing different types	Storing different types
Variable forms and unstructured data	Flat data models	Variable fields and unstructured data	Similar data types in a column	Complex and relative data models
Frequently read and write operations	For application with frequent write operations	For applications with frequent read and write operations	For applications with frequent read from different columns	—
Query in several fields	Querying by a key	Querying by every fields	Querying by limited numbers of columns	Querying by nodes

In the health sector, all kinds of data are seen such as medical findings, measurements, artifacts x-ray images, and electroencephalography wave recordings. A great amount of these data is document-based and unstructured. The main challenge is the frequent change of the form definitions; and the system must be able to be updated [31].

Considering the exponential growth in healthcare data, it is necessary to provide a resistant storage system for maintaining these data per second. On the other hand, each time, the system must be capable to retrieve every amount of data. So, custom operations in healthcare database systems are “write” and “read”. Querying on different fields is also necessary.

In the case of NoSQL models, column-based database is usable for a lot of small continuous reads and writes. The data are retrieved by specified pattern, and a few columns are involved in query. For accessing to high performance writing, key-value models are noticeable. This model also worked with a flat data model, and query is performed in defined keys. Document-based database operates over a wide variety of access patterns and data types. This database can handle numerous reads and writes. Document-based database supports complex structure and flexible number of columns without the need to build a schema again. This model also supports complex queries in different fields. Graph-based models are used to build relationships between objects such as social networks [32–34]. Table 2 contrasts health data characteristics with properties of different types of NoSQL DBs. Considering the comparison made in Table 2, the document-based database is the most adjacent database that can fulfill health data requirements including different types of data, variable forms, frequently read and write operations and etc. So, we use document-based database for our research in addition to the former model of data to make a comparison among them. The effect of data distribution is also verified in this study by using Shard’s utility in Cloud environment. Shard is a data management approach that has been exhibited in distributed databases.

4. Results and discussions

In this section, the performance of the proposed and previous model is discussed from the perspectives of query time in several operations, data preparation, flexibility and extensibility.

Table 3 – Parameters of the system for implementation.

System’s parameters	Value
Operating System	Windows 7 Ultimate, 64 bit
Processor	AMD Opteron processor / 2.30 GHz(2 processor)
Relational Database	SQL Server 2008
Non-Relational Database	Mongo DB 2.2.3-rco
HDD	80 GB
RAM	8.00 GB
Data	Drug’s data

4.1. *Test data and environment*

We used real patient data obtained from Medical Council for the implementation process. Previously, these textual data were in a relational database (SQL Server) including drugs and the information of their effects. Therefore, the former and proposed methods were evaluated in this research. In order to attain the benefits of Cloud Computing, we leased four-node cluster in a Cloud on High Performance Computing Research Center of Amirkabir University of Technology. This Cloud was implemented by vCloud Air platform, and we used an IaaS[1] service of Cloud. First, the old and proposed databases were implemented in one of these nodes, and some tests were performed. Then, non-relational database was implemented in shard technique by four nodes, and the test was repeated in new conditions. Table 3 investigates the details of these nodes and data.

In SQL Server, data were stored in some tables while Document-Based model stores data in collections with BSON format (a kind of text format). So, data must be converted to text for importing to Document-Based database. Converting tables to text is performed in different ways including the use of SQL to Text software, establishing of “result to file property” in SQL Server and converting via programming methods in NoSQL databases. In this study, we used Php Driver in MongoDB as a programming method and transferred data from SQL Server to MongoDB.

[1] Infrastructure as a service.

Table 4 – The variation of queries.

Query	Query description
I	To write health data (Drugs and their effects)
II	To retrieve results containing one textual value (“Drug-name = Co-amoxiclav”)
III	To retrieve results containing tow textual value (“Drug-name = Co-amoxiclav” and “Side effect = dizziness”)
IV	To retrieve results containing tow textual value and one numerical value (“Drug-name = Co-amoxiclav” , “Milligrams = 500” and “Side effect = dizziness”)

4.2. *Query time*

To evaluate the query time, four different queries are made with varied complexities (Table 4). The variation in query time, assuming the database sizes, is measured by querying 1000, 10,000, 100,000, and 1000,000 records, respectively. Here, SQL Server is compared with MongoDB (a document-based database) in non-distributed and distributed environments named Mongo and Mongo-sharded, sequentially. In Mongo-sharded’s status, data are distributed in four machines on the introduced Cloud. The results are given in Tables 5–8 based on different queries.

Fig. 1 represents the most tangible results. As shown in Fig. 1, in “write” operation (query I), Mongo is faster than SQL Server for all database sizes because of schema-less property of NoSQL DBs. When we used shard in this database, “write” speed reduced due to network overhead. Nevertheless, “write” in Mongo-sharded is also faster than SQL Server in all sizes of data.

In “read” operation (query II to IV), SQL Server is more efficient than document-based DBs with no shard property due to indexing technique in relational databases. By adding shard capability in document based database, the performance impressively increases and approximately reaches to SQL Server. Query II to IV display the same result in the retrieve operation. Therefore, the types of query and data sizes do not have significant effect on the results of retrieving as shown in Fig. 1(b)–(d).

Hence, Mongo-sharded is efficient in “read” and “write” operations. It is noticeable since these operations are most usual in healthcare systems. Nevertheless, SQL Server has a very poor throughput in “write” workload.

We follow Database’s manner when the size of database is increasing. The effect of the size is deleted by calculating the ratio of response time to numbers of records in different queries. The results are presented in Tables 9–11.

The response times of SQL Server and Mongo decrease slowly by growth of the database’s sizes (Tables 9 and 10). It is noticeable that these models are in a non-distributed environment, and SQL Server cannot act in distributed systems, as well. As shown in Table 11, the average of the response time decreases by increase in database’s size in the Mongo-shard model. It can be concluded that Mongo-shard is also capable to be involved with huge and distributed data as it was mentioned in previous sections.

Table 5 – Query performance of different databases with various sizes: query I.

Number of records	Response time(s) in various database implementations		
	SQL Server	Mongo	Mongo-sharded
1000	1.13	0.24	1.15
10,000	12.34	2.41	7.29
100,000	117.9	23.8	68.6
1,000,000	1161.31	248.46	700.49

Table 6 – Query performance of different databases with various sizes: query II.

Number of records	Response time(ms) in various database implementations		
	SQL Server	Mongo	Mongo-sharded
1000	1.8	1.39	5.37
10,000	8.05	11.23	3.96
100,000	40.89	110.56	28.6
1,000,000	386.52	1663.87	307.99

Table 7 – Query performance of different databases with various sizes: query III.

Number of records	Response time(ms) in various database implementations		
	SQL Server	Mongo	Mongo-sharded
1000	1.64	1.25	3.39
10,000	6.55	10.41	4.36
100,000	44.57	103.68	28.36
1,000,000	365.45	1444.43	269.7

Table 8 – Query performance of different databases with various sizes: query IV.

Number of records	Response time(ms) in various database implementations		
	SQL Server	Mongo	Mongo-sharded
1000	1.44	1.3	2.46
10,000	5.79	10.45	9.34
100,000	38.13	103.83	60
1,000,000	408.65	1111.06	336.33

4.3. *Data preparation*

Data preparation includes activities that prepare data for importing into databases such as the definition of tables, primary keys and foreign keys, the creation of relationships between tables, and data normalization. These processes are necessary for a relational database. But, NoSQL DBs is a schemaless DB, and row data are imported directly into databases with no need for fitting them. Furthermore, NoSQL DBs are more advantageous than traditional DBs since they can significantly reduce the database development time.

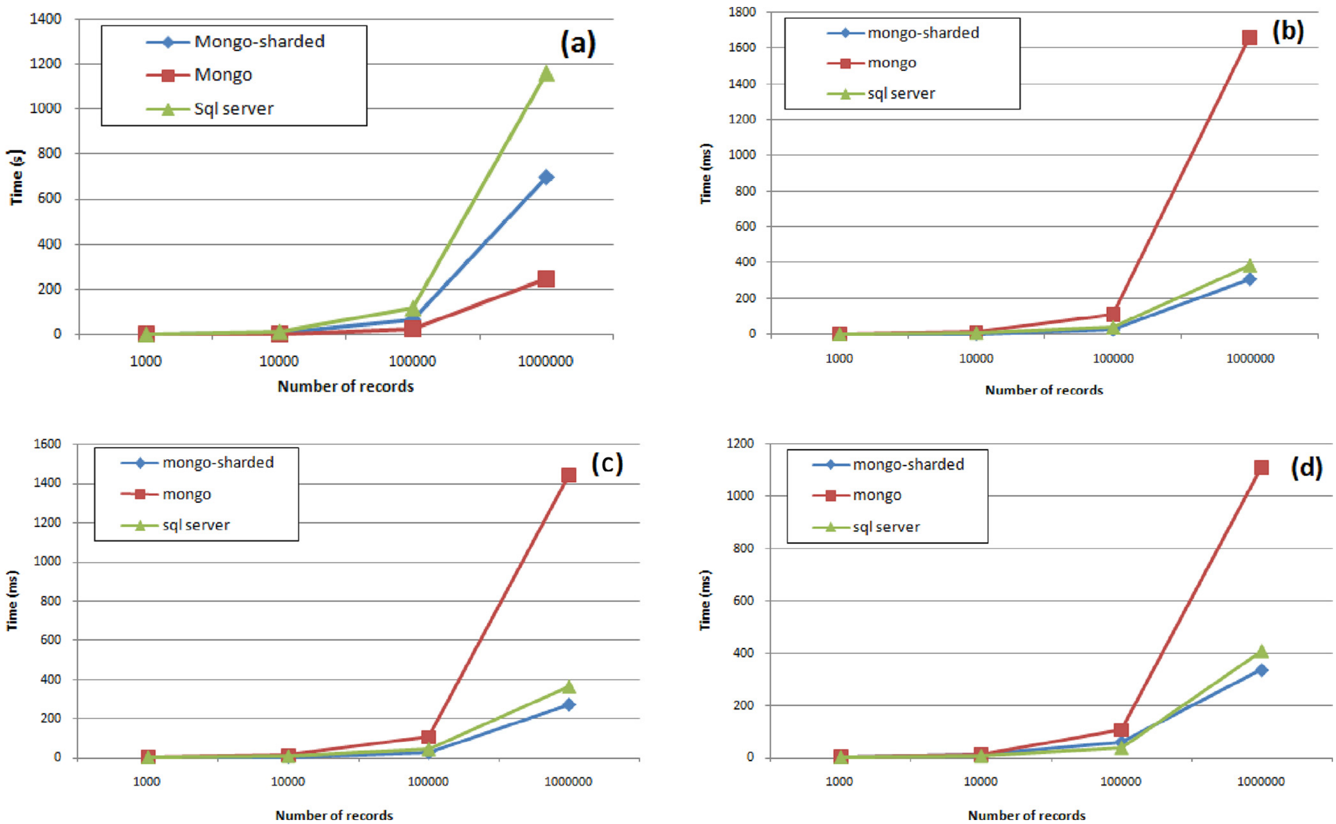

Fig. 1 – Variation in query time with DB size in different environments: (a) query I, (b) query II, (c) query III, (d) query IV.

4.4. *Flexibility*

Form definitions are changed frequently in a healthcare system. The system must be able to store data in new formats, as explained in previous sections. Traditional databases had a static nature and defined the initial forms at first. Data modeling is limited here by the permissible number of columns. Hence, the redefinition of the forms is a challenge in this data model. In contrast, document-based DBs have a flexible structure with no schema need. When the content changed, data can be added without re-designing of the schema. Hence, complex data are always simply inserted into these models.

4.5. *Extensibility*

According to the data growth in healthcare, it is necessary to deploy extensible models. Relational database is not planned for a huge amount of data. Also, they arduously work in a distributed environment. Thus, from the extension efficiency perspective, relational DBs are not proper model. In contrast, NoSQL DBs have been prepared potentially for a large amount of data in a distributed manner. Data can be spread straightforward on different machines in several geographical regions. Essentially, there are no limitations on data volume. Also, NoSQL DBs are applicatory in a Cloud environment that amplifies extensibility.

Table 9 – The ratio of SQL Server's response time (μs) to numbers of records in different queries.

Number of records	Query I	Query II	Query III	Query IV
1000	1130	1.8	1.64	1.44
10,000	1234	0.81	0.66	0.58
100,000	1179	0.41	0.45	0.38
1,000,000	1161	0.39	0.37	0.41

Table 10 – The ratio of Mongo's response time (μs) to numbers of records in different queries.

Number of records	Query I	Query II	Query III	Query IV
1000	240	1.8	1.64	1.44
10,000	241	0.81	0.66	0.58
100,000	238	0.41	0.45	0.38
1,000,000	248	0.39	0.37	0.41

Table 11 – The ratio of Mongo-sharded's response time (μs) to numbers of records in different queries.

Number of records	Query I	Query II	Query III	Query IV
1000	1150	5.37	3.39	2.46
10,000	729	0.4	0.44	0.93
100,000	686	0.29	0.28	0.6
1,000,000	700	0.31	0.27	0.34

5. Conclusion

In the present paper, considering the current challenges with respect to health data management, a model based on the Document-based databases and Cloud Computing was presented. In order to evaluate the performance of this model compared to previous one, these two models were implemented by using real health data. Assuming the query time, Document-based database acted much faster than SQL Server in the "Write" operation for different database sizes. In the case of "Read" operation for different queries and DB sizes, SQL Server acted better than Document-based DB because of its indexing property. By adding Shard feature to Document-based DB, the speed of data recovery operations highly increased and reached almost to SQL Server. However, Sharding affected the writing speed in this database and slowed down the "write" operations more than the previous state. Nevertheless, the "write" speed was still more than SQL Server.

Considering data-preparation, the new model has the capacity of storing semi-structured and unstructured data with no preparation need. The new model also has higher flexibility than the former model. The extensibility of the new model is high, while the former model has some limitations in this regard.

The current study provided a solution for storing and processing of health data. The assessments performed in different contexts indicate the effectiveness of the solution. In the future, this model may be developed through applying modern storage and processing tools such as map-reduce programming models. In this model, a considerable issue is trust in data security and privacy by users. To obtain this issue, various security approaches in Cloud environment must be investigated; and compatible method should be introduced for these conditions.

Acknowledgment

I would like to express my gratitude to the members of High Performance Computing Research Center (HPCRC) of Amirkabir University of Technology for their support and willingness to spend some time to contribute to this work.

REFERENCES

[1] Y. Pingle, V. Kohli, Sh. Kamat, N. Poladia, Big data processing using Apache Hadoop in cloud system, Int. J. Eng. Res. Appl. 2 (2012). 475–480.

[2] J. Roski, G.W. Bo-Linn, T.A. Andrews, Creating value in health care through big data: opportunities and policy implications, Health Aff. 33 (7) (2014).

[3] W.R.A.V. Raghupathi, Big data analytics in healthcare: promise and potential, Health Inf. Sci. Syst. 2 (3) (2014) 1–10.

[4] H.C. Koh, G. Tan, Data mining applications in healthcare, J. Healthc. Inf. Manag. 19 (2) (2011) 65.

[5] K.K.-Y. Lee, W.-C. Tang, K.-S. Choi, Alternatives to relational database: comparison of NoSQL and XML approaches for clinical data storage, Comput. Methods Programs Biomed. (2013) 99–109.

[6] Big Data, http://www.gartner.com/it-glossary/big-data, 2015.

[7] N. Leavitt, Will NoSQL Databases Live Up to Their Promise, IEEE Computer Society, 2010.

[8] S. Tiwari, Professional NoSql, John Wiley & Sons, Inc, Indiana, 2011.

[9] L.Q. Ji Qi, Z. Luo, Distributed structured database system HugeTable, in: Cloud Computing, First International Conference, Beijing, China, 2009.

[10] D. Bruhn, Comparison of Distribution Technologies in Different NoSQL Database Systems, Institute of Applied Informatics and Formal Description Methods, 2011.

[11] S. Sakr, A. Liu, D.M. Batista, M. Alomari, A survey of large scale data management approaches in cloud environments, IEEE Commun. Surv. Tutor. 13 (3) (2011).

[12] A.R. Syed, K. Gillela, C. Venugopal, The future revolution on big data, IJARCCE 2 (6) (2013).

[13] S. Bushik, Vendor-Independent Comparison of NoSQL Database: Cassandra, HBase, MongoDB, Riak, Altoros System, 2012.

[14] N.C. Brad, Data De-Duplication in No SQL DB, 2012.

[15] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Fut. Gener. Comput. Syst. 25 (6) (2009) 599–616.

[16] N. Lewis, Poor data management costs healthcare providers, in: InformationWeek Healthcare, 2012.

[17] L. Wang, R. Ranjan, J. Kolodziej, A. Zomaya, L. Alem, Software tools and techniques for big data computing in healthcare clouds, Fut. Gener. Comput. Syst. 43 (2015) 38–39.

[18] W.-S. Li, J.Y. Ying Yan, J. Zhang, Xbase: cloud-enabled information appliance for healthcare, in: 13th International Conference on Extending Database Technology, Lausanne, Switzerland, 2010.

[19] Mphrx. http://mphrx.com/.

[20] Dossia. http://www.dossia.org/.

[21] Microsoft healthvault. https://www.healthvault.com/ir/en, 2015.

[22] Google Health. https://www.google.com/intl/en_us/health/about/, 2015.

[23] A. O'Driscolla, J.D. Roy, D. Sleator, "Big data", Hadoop and cloud computing in genomics, J. Biomed. Inform. 49 (5) (2013) 774–781.

[24] V. Nguyen, R. Wynden, Y. Sun, HBase, MapReduce, and Integrated Data Visualization for clinical signal data, in: Computational Physiology, 2011. Papers from the 2011 AAAI Spring Symposium: Stanford, California, USA.

[25] C. Doukas, T. Pliakas, I. Maglogiannis, Mobile healthcare information management utilizing Cloud Computing and Android OS, in: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE, 2010.

[26] S. Van Poucke, Z. Zhang, M. Schmitz, M. Vukicevic, M.V. Laenen, L.A. Celi, et al., Scalable predictive analysis in critically ill patients using a visual open data analysis platform, PLoS ONE 11 (1) (2016) e0145791.

[27] S. Oh, J. Cha, M. Ji, H. Kang, S. Kim, E. Heo, et al., Architecture design of healthcare software-as-a-service platform for cloud-based clinical decision support service, HIR 21 (2) (2015) 102–110.

[28] M. Vukićević, S. Radovanović, M. Milovanović, M. Minović, Cloud based metalearning system for predictive modeling of biomedical data, Scientific World Journal 2014 (2014).

[29] M. Parekh, B. Saleena, Designing a cloud based framework for healthcare system and applying clustering techniques for region wise diagnosis, Procedia Comput. Sci. 50 (2015) 537–542.

[30] I. Robinson, J.W. Emil Eifrem, N.J. Mike Loukides (Ed.), Graph Databases, O Reilly media, United States of America, 2013.
[31] O. Schmitt, T.A. Majchrzak, Using document-based databases for medical information systems in unreliable environments, in: 9th International ISCRAM Conference, 2012.
[32] Difference between Document-based and Key/Value-based databases? http://stackoverflow.com/questions/3554169/difference-between-document-based-and-key-value-based-databases, 2013.
[33] What is a Key/Value store database? http://dba.stackexchange.com/questions/607/what-is-a-key-value-store-database.
[34] 35+ Use Cases for Choosing Your Next NoSQL Database. highscalability.com/blog/2011/6/20/35-use-cases-for-choosing-your-next-nosql-database.html, 2011 (accessed 05.05.15).

Zohreh Goli-Malekabadi *
Morteza Sargolzaei-Javan
Mohammad Kazem Akbari
Cloud Research Center (CRC), Amirkabir University of Technology, Tehran, Iran

* *Corresponding author.* Tel.: +98 913 289 8330.
E-mail address: zh.goli22@gmail.com (Z. Goli-Malekabady).

http://dx.doi.org/10.1016/j.cmpb.2016.04.016