VNU Journal of Science: Policy and Management Studies, Vol. 33, No. 2 (2017) 146-156

Big Data System for Health Care Records

Phan Tan[1], Nguyen Thanh Tung[2,*], Vu Khanh Hoan[3],
Tran Viet Trung[1], Nguyen Huu Duc[1]

[1]*Institute of Information Technology and Communication, Hanoi University of Science and Technology, 1 Dai Co Viet Street, Hai Ba Trung, Hanoi, Vietnam*

[2]*VNU International School, Building G7-G8, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*

[3]*Nguyen Tat Thanh University, 300A, Nguyen Tat Thanh, Ward 13, District 4, Ho Chi Minh City, Vietnam*

Received 12 April 2017
Revised 12 May 2017; Accepted 28 June 2017

Abstract: So far, medical data have been used to serve the need of people's healthcare. In some countries, in recent years, a lot of hospitals have altered the conventional paper medical records into electronic health records. The data in these records grow continuously in real time, which generates a large number of medical data available for physicians, researchers, and patients in need. Systems of electronic health records share a common feature that they are all constituted from open sources for Big Data with distributed structure in order to collect, store, exploit, and use medical data to track down, prevent, treat human's diseases, and even forecast dangerous epidemics.

Keywords: Epidemiology, Big data, real-time, distributed database.

1. Introduction

So far, medical data have been used to serve the need of people's healthcare. Big Data is an analytic tool currently employed in many different industries and plays a particularly important role in medical area. Medical health records (or digitalized) help produce a big database source which contains every information about the patients, their pathologies and tests (scan, X-ray, etc.), or details transmitted from biomedical devices which are attached directly to the patients.

In many countries worldwide, health record systems have been digitalized on national scale, and this data warehouse has contributed greatly to improving patients' safety, updating new treatment methods, helping healthcare services get access to patients' health records, facilitating disease diagnoses, and developing particular treatment methods for each patient basing on genetic and physiological information. Besides, this data warehouse is a big aid for disease diagnosis and disease early warning, especially for the most common fatal ones worldwide such as heart diseases and ovarian cancer, which are normally difficult to detect.

In healthcare, Big Data can assist in identifying patients' regimens, exercises,

*Corresponding author. Tel.: 84-962988600.
Email: tungnt@isvn.vn

https://doi.org/10.25073/2588-1116/vnupam.4101

preventive healthcare measures, and lifestyle aspects, therefrom physicians will be able to compile statistics and draw conclusion about patients' health status. Big Data analysis can also help determine more effective clinical treatment methods and public health intervention, which can hardly be recognized using fragmented conventional data storage. Medical warning practice is the latest application of Big Data in this area. The system provides a profound insight into health status and genetic information, which allows physicians to make better diagnoses of disease's progress and patient's adaptation to treatment methods.

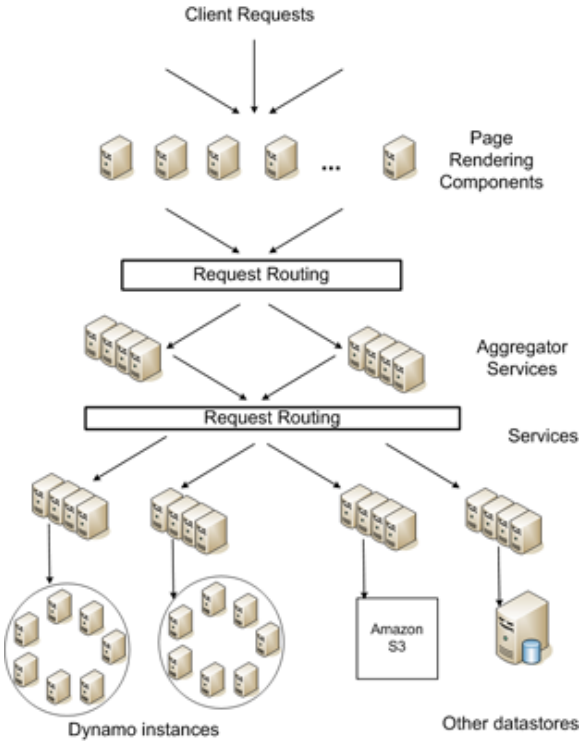
In Vietnam, using Big Data systems to collect, store, list, search, and analyze medical information to identify diseases and epidemics is a subject that attracts much attention from researchers. Among those systems is HealthDL.

Health DL, a system distributing, collecting, and storing medical Big Data, is constructed optimally for data received from health record history and biomedical devices which are geographically distributed with constant increase in real-time.

The next part of this article consists of the following main contents: (1) introducing related researches, (2) analyzing and describing input data characteristics of the HealthDL, (3) designing a general system model, integrating system components, (4) discussing experimental results, and efficiency evaluation. The last part summarizes our work and opens for future study.

2. Related work

According to [1], in conventional electronic health record systems, data are stored as tuples in relational database tables. The article also indicates that the use of conventional database systems is facing challenges relating to the availability due to the quick expansion of the throughput in healthcare services, which leads to a bottleneck in storing and retrieving data. Moreover, in [2], the writers show that the variety of increasing medical data together with the development of technology, data from sensor, mobile, test images, etc. requires further study into a more suitable method to organize and store medical data.



Picture 1. Dynamo Amazon Architecture.

Researches [1] and [3] point out the necessary requirements of Electronic Health Record (EHR) and suggest using non relational database model (NoSQL [4]) as a solution to storing and processing medical Big Data. However, [1] and [3] only propose a general approach but not introduce an overall design including collecting and storing EHR. These researches are also executed without experiment, installation and evaluation on the efficiency of the system. Among NoSQL solution, Document-oriented database is widely expected as the key to health record storage, which includes patients' records, research reports, laboratory reports, hospital records, X-ray and CT scan image reports, etc.

The writers [1] suggest using Dynamo Amazon [5], an Amazon cloud database service, to store constant data streams sent from biomedical devices. Amazon Dynamo architecture relies on consistent hashing for open mechanism and uses virtual nodes to distribute data evenly on physical nodes and vector clock [6] to resolve conflicts among data versions after concurrency.

Apart from data storage components under NoSQL model as stated in related studies, HealthDL, a general system, also integrates distributed message awaiting queues to collect data from geologically distributed biomedical devices. Experimental results are mentioned in part 5.

3. Medical data sources of the system

Medical data referred to in this study belong to two main groups: data collected from patients' records and data transmitted from biomedical devices. Below is the data input description of HealthDL system.

Health record data

Data analyzed are collected from four groups of diseases below:

- Hypertension: tuple dimension from 800-1000 bytes

- Pulmonary tuberculosis: tuple dimension from 400-600 bytes

- Bronchial asthma: tuple dimension from 500-700 bytes

- Diabetes: tuple dimension from 800-1000 bytes

The typical characteristic of health record data is its flexibility. Each type of disease composes of different data amounts and domains. For hypertension, each record document contains about 75 separate domains whose structures are split into 3 or 4 layers. This number of layers is 4 or 5 for the other three groups of diseases.

Data from biomedical devices

Patients' data are transmitted continuously from multiindex biomedical monitors to the system in the real-time of once every second. If 1000 patients are observed by independent monitors within one month, each patient is examined for 2 hours per day, the information received from biomedical devices will be 216.000.000 packages of data. If each package contains 540 bytes, the information coming from biomedical devices will reach a huge amount of about 116 Gigabytes.

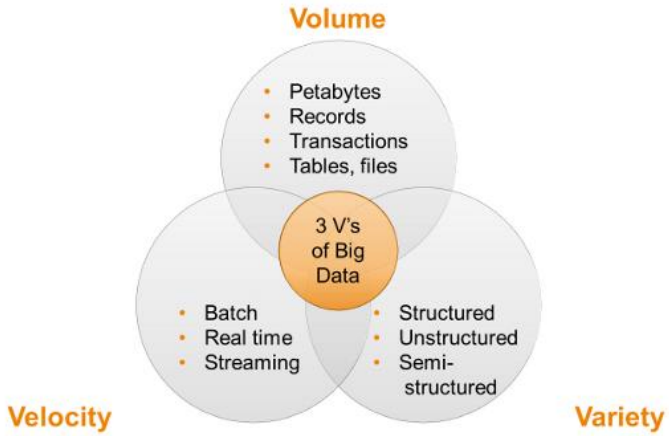
4. Characteristics of medical data in HealthDL system

- **Big Volume**: as mentioned above, the amount of data received within a month when monitoring 1000 patients with independent monitors is 116 gigabytes. As a result, when the number of patients increases, the amount of data will be extremely enormous.
- **Big Velocity**: data are generated continuously from biomedical devices at high speed (one tuple per second), which requires high speed of data processing (reading and writing). Moreover, when the speed of generating data becomes higher and higher, the speed of storing and processing data must be compatible with input data in real-time.
- **Big Variety**: with the outburst of internet devices, data sources are getting more and more diverse. Data exist in three types: structured, unstructured, and semi-structured. Medical records belong to semi-structured data with irregular schema.
- **Big Validity**: medical data are stored and utilized aiming at high efficiency in disease diagnoses and treatment, as well as epidemic warning, which partly improves health checkup, disease treatment quality, and reduces test fees.

Comment: medical data source in HealthDL carries the typical feature of Big Data.

Big Data is a terminology used to indicate the processing of such a big and complex data set that all conventional data processing tools cannot meet its requirements. These requirements include analyzing, collecting, monitoring, searching, sharing, storing, transmitting, visualizing, retrieving and assuring the privacy of data.

Big Data contains a lot of precious information which, if extracted successfully, will be a great help for businesses, scientific studies, or warning of potential epidemics relying on the data it collects.



Picture 2. 3 V's of Big Data.

System Model

We constitute HealthDL system with the overall structure divided into four main blocks as followed:

1. The component block of biomedical devices measuring essential indices from patients

2. The component block of receiving and transmitting data

3. The component block of storing health records

4. The component block of storing data received from biomedical devices

The input of the system includes two major streams:

1. Input data of health records stored in specific databases, which are optimized for health record data with flexible structure.

2. Input data coming from biomedical devices, which goes through a waiting queue and then stored in a database.

5. Suggested technology

MongoDB for storing health record data

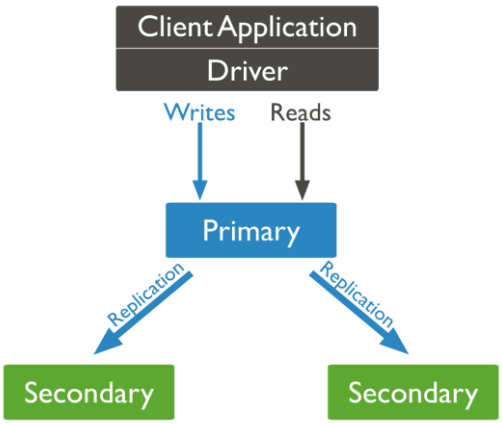
MongoDB [7] is a NoSQL document-oriented database written in C++. Consequently, it possesses the ability to calculate at high speed and some outstanding features as followed:

- **The Model of flexible data**: MongoDB does not require users to define beforehand database schema or structures of stored documents, but allows immediate changes at the time each tuple is created. The data

is stored in tuples using JSON like format with flexible structures.

- **High scalability**: allowing the execution in many database centers: MongoDB can expand in one data centre or be implemented in many geologically distributed data centers.
- **High availability**: MongoDB possesses a good ability to balance the load and integrate data managing technologies when the size and throughput of data rise without delaying or restarting the system.
- **Data Analysis**: MongoDB database supports and supplies standardized control programs to integrate with analyzing, performing, searching, and processing spatial data schema.
- **Replication**: this important feature of MongoDB permits the duplication of the data to a group of several servers. Among those servers, one is primary and the rest are secondary. The primary replication server is in charge of general management, through which all manipulation and data updating are administered. Secondary servers can be employed to read data so as to balance load. MongoDB runs with automatic failover. Therefore, if the primary replication server happens to be unavailable, one of the secondary servers will be allowed to become the primary server to assure the success of data writing.

Designed as document-oriented database, MongoDB is the most suitable to store health record data with a vast number of domains, irregular domains, or of different patients. Its document-oriented structure allows users to create indexes for the quick search of health record information basing on text characteristics.



Picture 3. Replication in MongoDB.

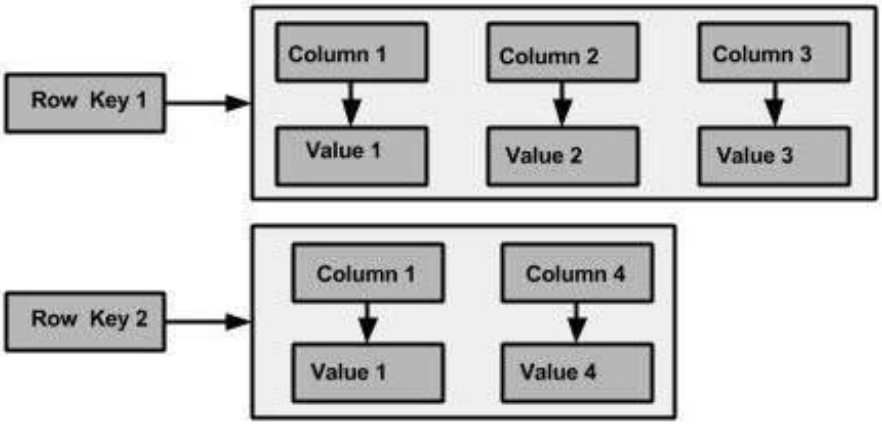
Cassandra database for data from biomedical devices

Cassandra [8] is an Apache open source distributed database with high scalability and based on peer-to-peer [9] architecture. In this system, all server nodes play equal roles; therefore, no component in this system is bottleneck. With remarkable fault-tolerance and high availability, Cassandra can organize a great amount of structured data.

- **Customizability and scalability**: as an open source software, Cassandra allows users to make any addition to primary server to meet their load demand and simultaneously permits partial withdrawal or complete move from primary server to reduce power consumption, replace, restore, and recover from errors without interrupting or restarting the system.
- **Architecture of high availability**: nodes in primary servers in Cassandra system are independent and are connected to other nodes within the system. When one single node fails to perform correctly and stops working, data reading manipulations can be processed by other ones. This

mechanism assures the smooth operation of the system.

- **Elastic data model**: Cassandra database system is designed bearing column-oriented model which allows the storage of structured, unstructured, as well as semi-structured data (picture 4) without having to define beforehand the data schema as in the case of relational data.
- **Easily distributed data**: Cassandra organizes primary nodes into clusters in round format and uses consistent-hashing [10] to distribute data, which maximizes data transmission competence when the system's configuration changes. Any primary nodes added or moved will have no effects on the redistribution of the data space.
- **Quick data writing with big throughput**: Although Cassandra is designed to run on common computers with low configuration, it is capable of achieving high efficiency, reading and writing big throughput, and storing hundreds of terabytes without reducing the efficiency of data reading and processing.



Picture 4. Cassandra column-oriented Model.

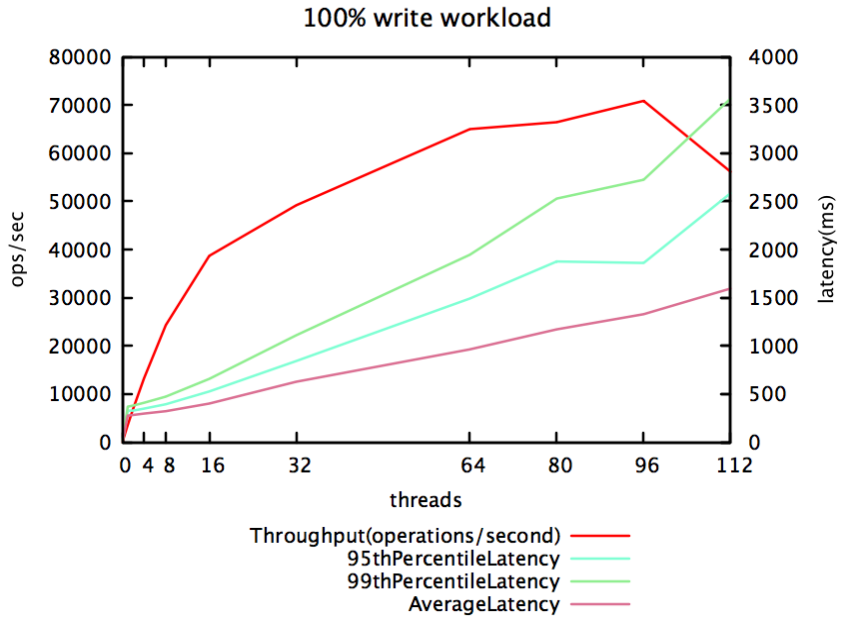
6. Experimental evaluation

In this part, we assess the efficiency of HealthDL system in reading and writing data in distributed environment when connection concurrencies accelerate. We installed and carried out experimental running on MongoDB and Cassandra using two standard evaluation tools including YCSB [11] and Cassandra-stress [12].

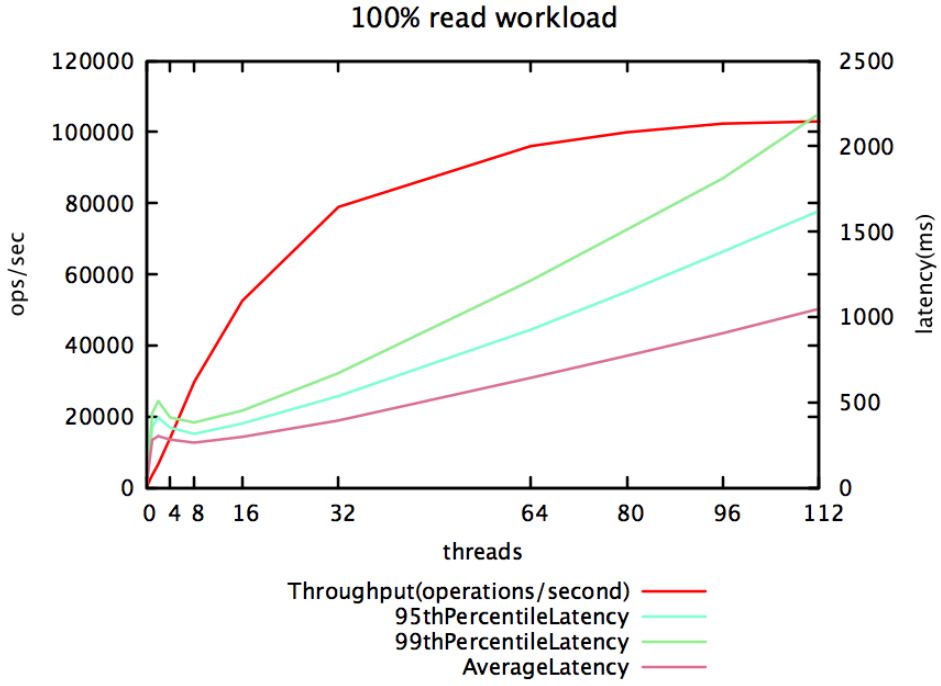
Evaluation on MongoDB component for storing health record data

MongoDB is installed in virtualized environment using docker-compose [13], a computer cluster consisting of 30 virtual nodes sharing the configuration as followed: CPU: 02 x Haswell2.3G, SSD: 01 Intel 800GB SATA 6Gb/s, RAM: 128GB.

The result for the scenario of solely reading and writing data reveals high efficiency, with writing and reading speed reported from 70000 to 100000 operations per second, the latency recorded from 1s to 1.5s with 1 to 100 client concurrencies. (Picture 5, 6).

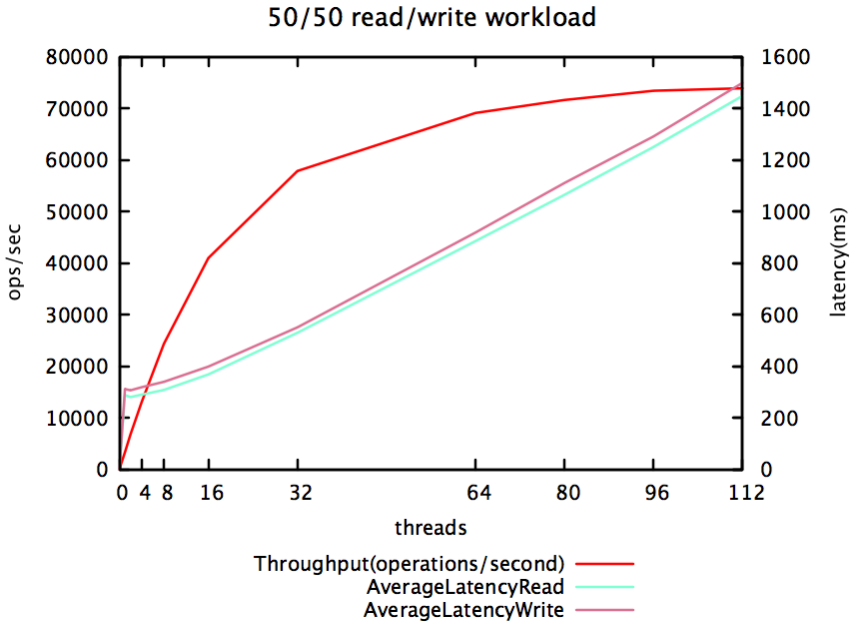


Picture 5. Scenario of writing data in MongoDB.

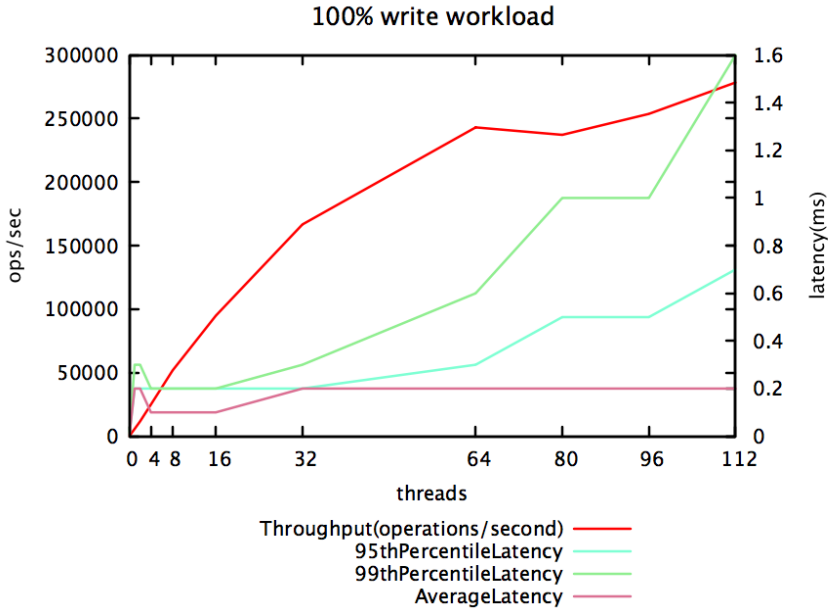


Picture 6. Scenario of reading data in MongoDB.

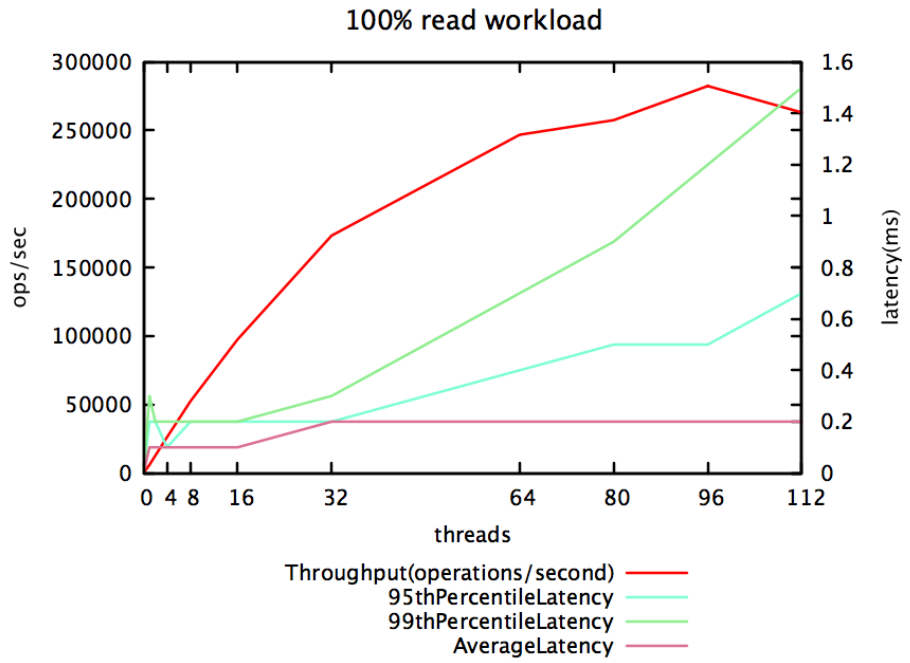
The scenario of reading and writing at the proportion of 50/50 (simultaneous reading and writing) also shows positive signs, with the average speed of 70000 operations per second and the average latency marked at 1.4s (Picture 7).



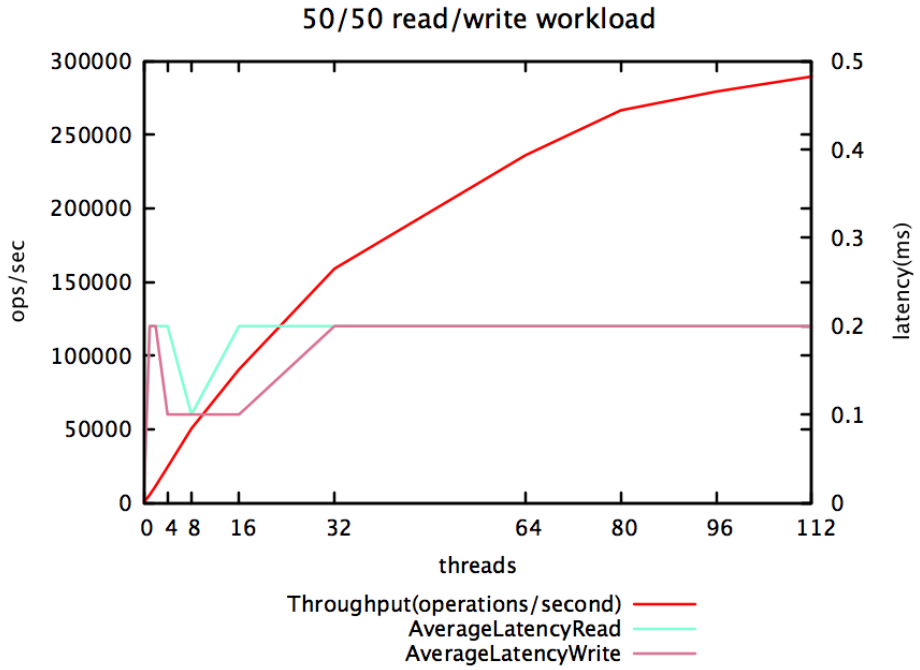
Picture 7. The scenario of concurrent reading and writing in MongoDB.



Picture 8. Increase in concurrent writing operations.



Picture 9: Increase in concurrent reading operations.



Picture 10. Simultaneous reading and writing operations in Cassandra.

Evaluation on Cassandra component for storing data received from biomedical devices

Cassandra was installed in 3 separate servers with the configuration of each one as followed: CPU: 02 x Haswell 2.3G, SSD: 01 Intel 800GB SATA 6Gb/s, RAM: 128GB. In the experimental scenario, the number of reading and writing operations per second and the average latency were calculated. Experiments revealed the increase in the number of clients executing reading and writing data in concurrency. In the experiment where concurrencies only executed writing operations (picture 9), Cassandra showed high efficiency with 250000 to 300000 operations per second. The average latency is 0.2 to 0.3 ms. For simultaneous reading and writing scenario (picture 10), Cassandra still responded with 250000 to 300000 operations per second.

Experimental outcomes executed in MongoDB and Cassandra in concurrent environment indicates that their components produces high efficiency even under the circumstance of reading and writing concurrently. Cassandra supports a higher number of operations per second. Consequently, it presents greater suitability for storing medical data collected from real-time biomedical devices.

7. Conclusion

In this article, we have introduced a system for collecting and storing medical data named HealthDL. The results relating to the efficiency of storing components in experimental environment have proved its high possibility to meet the professional requirements of reading and writing concurrent data. As for overall design, the system is constituted from distributed components with high customizability and elastic data support. In the future, we will apply this system and integrate it with other components for analyzing distributed medical data.

Acknowledgements

The writers of this article would like to send sincere thanks to National Scientific Study Program, which aims at stable development in the Northwest, for its sponsor to this scientific subject "Applying and Promoting System of Integrated Softwares and Connecting Biomedical Devices with Communications Network to Support Healthcare Delivery and Public Health Epidemiology in the Northwest" (Code number: *KHCN-TB.06C/13-18)*

References

[1] M. Z. Ercan and M. Lane, "An evaluation of NoSQL databases for EHR systems," in Proceedings of the 25th Australasian Conference on Information Systems, 2014, pp. 8–10.

[2] J. Andreu-Perez, C. C. Y. Poon, R. D. Merrifield, S. T. C. Wong, and G.-Z. Yang, "Big data for health," IEEE J. Biomed. Heal. informatics, vol. 19, no. 4, pp. 1193–1208, 2015.

[3] C. Dobre and F. Xhafa, "NoSQL Technologies for Real Time (Patient) Monitoring," in Advanced Technological Solutions for E-Health and Dementia Patient Monitoring, IGI Global, 2015, pp. 183–210.

[4] K. Grolinger, W. a Higashino, A. Tiwari, and M. A. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores," J. Cloud Comput. Adv. Syst. Appl., vol. 2, p. 22, 2013.

[5] G. Decandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: amazon's highly available key-value store," ACM SIGOPS Oper. Syst. Rev., vol. 41, no. 6, p. 220, 2007.

[6] D. S. Parker, G. J. Popek, G. Rudisin, A. Stoughton, B. J. Walker, E. Walton, J. M. Chow, D. Edwards, S. Kiser, and C. Kline, "Detection of Mutual Inconsistency in Distributed Systems," IEEE Trans. Softw. Eng., vol. SE-9, no. 3, pp. 240–247, May 1983.

[7] K. Chodorow, MongoDB: the definitive guide. " O'Reilly Media, Inc.," 2013.

[8] A. Lakshman and P. Malik, "Cassandra: a decentralized structured storage system," ACM SIGOPS Oper. Syst. Rev., vol. 44, no. 2, pp. 35–40, 2010.

[9] S. Androutsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” ACM Comput. Surv., vol. 36, no. 4, pp. 335–371, Dec. 2004.

[10] D. Kargerl, T. Leightonl, and D. Lewinl, “Consistent Hashing and Random Trees : Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web,” Most, pp. 654–663.

[11] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, “Benchmarking Cloud Serving Systems with YCSB,” System.

[12] “Cassandra Stress.” [Online]. Available: http://docs.datastax.com/en/cassandra/2.1/cassandra/tools/toolsCStress_t.html.

[13] “Docker.” [Online]. Available: https://docs.docker.com/engine/docker-overview/.

"Docker Compose" [Online]. Available: https://docs.docker.com/compose/overview/.