Distinct Features of Healthcare Systems and An Architecture to Manage Healthcare Distributed Databases

Subrata Kumar Das
Dept. of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

Mohammad Zahidur Rahman
Dept. of Computer Science and Engineering
Jahangirnagar University
Savar, Dhaka, Bangladesh

ABSTRACT

Healthcare organizations store patient records (EMR, EHR, and PHR) in electronic formats to recall urgent information about the patients. The sharing of the health record is very much important to make urgent decisions about the individuals care and quality treatment. Unlike other database systems, healthcare repository systems have distinct features. Data accesses here are localized most, not performed for all patients equally and from different locations at a time, etc. As a result, managing the distributed databases containing health data using a traditional database management system (DBMS) is not a better choice. This article proposes an architecture to manage nodes by keeping transparency and share data by identifying the distinctive features of the healthcare systems. As the volume of the patient data is increasing hugely in the medical sector, other pressing issues are reducing memory consumption and managing replication to ensure data reliability. This article also proposes a new replication strategy by relating to the proposed architecture to reduce the number of replicas, network overhead, and memory consumption without compromising performance. The experiment result shows that the proposed system outperforms the existing conventional systems.

Keywords

Health Records, DDBMS, Architecture, ID Index Server, Area-based Replication

1. INTRODUCTION

Healthcare data collected from a patient for treatment purposes by a physician are stored in electronic format to use later easily. The healthcare databases contain three types of electronic health records (EMR, EHR, and PHR) and scatter in different locations. Electronic Medical Records (EMR) include pharmaceutical records, clinical laboratory results, patient diagnosis, disease progression, etc. and managed by the specialists of a single healthcare organization. Electronic Health Records (EHR) are information that consists of healthcare data from different organizations managed by the provider. Personal Health Records (PHR) are the information that is created by the patient and gathered by themselves from physicians, labs, etc. to track their health information. The sharing of the health record is very much important to make urgent decisions about the individuals care and quality treatment [4] [6]. The sharing of health data from different sources also reduces the treatment cost significantly due to the availability of previous reports.

However, the existing systems that use traditional database management system (DBMS) to interconnect distributed databases, especially healthcare databases, and share data among them could not provide a better solution. In existing database management systems, each site keeps a local DBMS. Each site also keeps a copy of the distributed DBMS and its associated global data directory. The data directory consists of the location of all data and the database related information. Each request from users is first sent to the associated global data directory to determine whether the transaction is local or global [15]. This strategy reduces the performance of the large-scale data management system because the global data directory here contains huge data that is checked for each user request.

Another pressing issue is the volume of patient data increasing hugely in the medical sector. So, the number of replicas should be small to reduce memory consumption. The traffic numbers among replicas should be also small as much as possible to decrease the concurrency management time and the network overhead.

Moreover, the owner of the health organizations is not a single one. So, health organizations would require a contract over the world for authorization to store replicas in different nodes. But, who will come forward for an initiative to contract globally, and how is it possible? These questions lead to a bunch of problems mentioned below.

(1) Some healthcare organizations could have a huge amount of data that needs a large volume of storage, while others could have very few data. So, all organizations would not contribute equally to develop the healthcare system.

(2) There are many small organizations that have no enough resources to store replicas according to existing replication methods. Adaptive replication strategy, one of the existing techniques, can make replicas considering the size discussed in section 4.4. However, big organizations would not agree to give space to small organizations for creating replicas globally without any negotiation or interest.

(3) As per existing replication procedures, small organizations have to update their system. Say, some organizations may able

to update their system as a world standard, but others may not agree to change their system that would make a barrier to share data from small organizations.

(4) Some organizations may wish to give a proportion of amount to worldwide big healthcare organizations to keep their replicas, but who would take responsibility to measure that proportion. Ultimately, small organizations could want to keep them back from such hazards.

This issue could be solved by limiting replication creation among localized scattered databases.

As a result, a generic architecture is highly needed to manage the healthcare distributed databases in a different strategy because the nature of health systems is different from other systems in terms of data accesses. The healthcare systems have some distinct features. Data accesses here are localized most, and not performed for all patients equally and from different locations at a time, etc. detailed in section 3.

To solve the above issues, in this study, we propose an architecture to interconnect nodes keeping transparency and manage data by considering the distinctive features of the healthcare systems. The proposed architecture could open a way to interconnect nodes of the patient databases scattered in different places. Users can access data from anywhere in the world after registering them under an organization.

This article also proposes a new replication strategy limiting the replica creation among localized databases to reduce the number of replicas, network overhead, and memory consumption.

Finally, the NS3 simulator is used to check the concurrency control management time, network overhead, and memory consumption of various existing replication strategies as well as our proposed replication method. The proposed system performs better than the existing conventional systems.

The remaining of the article is organized as follows. Section 2 presents significant related works. Section 3 describes the distinct features of the healthcare system and design considerations. Section 4 summarizes the existing replication strategies and their problems. Section 5 details the proposed architecture and Area-based Replication (ARB) technique. Section 6 explains the methodology of the experiment. Section 7 analyzes the data found from the simulator and shows the comparison among the proposed system and existing ones. Section 8 presents the limitation and solution. Finally, section 9 includes a conclusion summarizing the work.

2. RELATED WORK

The researchers are working to manage databases of health organizations globally for sharing and accessing data from remote places. OmniPHR [23] proposes a system to distribute health records using a well-known Chord algorithm. The main problem of OmniPHR is that the data have to be located in the model-enabled paths and must match with the standard supported by this model. Harika [7] provides a model to integrate health records using Blockchain technology. Kumari [12] also presents an architectural model using Blockchain integrating with wireless body area network (WBAN) to transfer the health data among different users. However, the application of each node must be updated in Blockchain and it doesnt allow redesign employing a strict logic. An E-health framework of the healthcare ecosystem is proposed to inter-operate patients data among individuals or consumers [5]. But this system is only focused on the existing E-health system of China. A P2P-based open health cloud is introduced to manage medicine and integrate various health data [2]. This system uses a master node like a NameNode of Hadoop. However, the whole system goes down when the master node crashes as the slave nodes are managed from here [17]. Besides, the data duplication may corrupt the data and arise noise that will prevent the correct services later. Shen et al. [24] propose a data-sharing scheme, MedChain, that leverages two separate networks: a peer-to-peer network for storage and a blockchain network. But this architecture is designed based on structured P2P network techniques and its responsiveness time increases with increasing numbers of directory nodes. Moreover, considering the various problems, several researchers try to tackle replication issues and develop different solutions. Some of the strategies are very simple, while others are complex to design, all proving different levels of efficiency. Most of the replications of data are supported by the systems, in which data is replicated at a node even though the node has not requested the data [25].

Awang et al. [1] focus on an Affinity Replica Selection Mechanism (ARSM) based on correlated or affine files that have common features accessed by different users. However, the patients' data are individual for each and can not access by different users at a time. Some articles [16] [13] [27] cite the random replica placement methods by placing replicas randomly without considering the necessity of data access. Those methods lead to an unbalanced load of data on sites and store replicas in a long network distance that affects the system data transmission performance [14]. OmniPHR [23] use Distributed Hash Tables (DHTs) for replication, which is structured overlays [18] [8] [10] [9]. Though the path replication strategy improves data availability, it may create an imbalance load and can not guarantee a fixed number of replicas [20]. Qi et al. [20] also provide a replication idea based on dividing the network into different zones, which does not also bring benefits for healthcare data for its access dominant locally. Purnama and Ashari [19] describe a concept of the distributed system for patient data where the replication creates in different nodes when patients signing up from those nodes like owner replication strategy. The number of replicas here increases in proportion to the number of signing up from different nodes. The problem is that patient may visit physically for authentication to access data from another node.

3. DISTINCT FEATURES OF THE HEALTHCARE SYSTEM

The healthcare database management system should be different by considering data access from other traditional distributed database management systems. Normally this database is highly personal; on the other hand, doctors need patient records in case of emergency.

3.1 Distinct Features

The main distinguishing features are mentioned below:

Local or area-based access rather than global Data accesses perform from the local database or Data accesses limit area basis on a group of databases in most cases; because the major patients take consultancy from local or nearer hospitals surrounding their city. In very few cases, data are accessed by users globally when the patient goes outside.

Data access not frequent for all patients All patients require to register their ID to the database to keep their information. Many patients need to go frequently to the doctor due to their physical severity, but others go for one or two times for general illness. So data against each ID would not access equally for all patients.

Data access not at a time There is less possibility to access data from different places at a time, as the patient can take treatment from one place at a moment. So, globally replication placement is not a great issue in the healthcare system, which arises inconsistency, consumes more disk spaces, expensive concurrency control, and network overhead.

3.2 Design Consideration

Consideration of distinct features of healthcare data provides some flexibility to design a new system. The distributed systems use replication to ensure data availability, but it also arises drawbacks like inconsistency including extra memory space. Therefore, a kind of system that could reduce the replicas as much possible can mitigate inconsistency and network overload as well as memory loss.

The distinct features of the healthcare system discussed in section 3 give support to design an architecture avoiding the replica creation globally for sharing medical data described below:

—As less data is accessed globally in the health care system, fewer data move over the networks due to access data stored in localized servers most of the time. So, the replication of records is not urgent globally in healthcare systems, which arises inconsistency and consumes huge memory space.

—Making ID Index differently for localized patients and global patients allows users to access data swiftly by identifying database information. Also, location transparency could be managed automatically by maintaining location addresses where data stored against patient ID.

—As all patients do not need regular consultancy, only those who are in severe illness go frequently for treatment, so data search could be fast by keeping frequent access data on top of the indexing.

—As one patient generally takes consultancy from one physician at a time, so it would not require partition tolerance according to the CAP theorem to support multiple accesses. However, the feature of partition tolerance to a system arises inconsistency.

—According to the CAP theorem, if many replicas are created of data to ensure availability, consistency management among more nodes takes much time that increases latency. As most are local access, the replication could be maintained among neighbor nodes by grouping on a small scale to ensure data availability that provides data access at low latency.

—A great challenge of data replication is managing updates with a low cost when assuring system consistency globally. Avoiding data replication globally could support to update data at a low cost. Here an Index Servers could be designed by taking the list of patient IDs with only location addresses of data, which could help to manage consistency easily by avoiding global data replication.

Therefore, we could consider the flexible points to reduce network overhead, save memory space, and control inconsistency. We could also take into account those to give location transparency, avoid partition tolerance, search swiftly through indexing with low latency.

4. EXISTING REPLICATION STRATEGY AND ITS PROBLEMS

The replication strategy is a process to create identical copies of data and store these copies at different distributed repositories or sites for ensuring data availability and improving data access performance. There are some exiting replication strategies used to ensure data reliability in the distributed environment. However, the existing techniques arise drawbacks, such as inconsistency control, huge memory consumption, network overhead, etc. for their large size of replicas and placement in nodes.

We below describe some existing methods used mostly to create the replica:

4.1 Owner Replication

The owner replication strategy creates the replicas of data only at the requesting node [16]. As per the technique, the replica surges in number in proportion to the increasing number of data requests. Besides, this technique takes a lot of time to propagate replicas over the network.

4.2 Path Replication

This strategy replicates the data on all nodes along the data transmission path from the source node contained data to the requesting node [16]. The path replication method is used in many distributed systems for its easy implementation and good search performance. However, it creates a very large number of replicas that make system inconsistency and waste more memory spaces.

4.3 Path Random Replication

The Path Random Replication is introduced as the number of replicas created by Path Replication can become very large eventually than necessary to achieve the required performance and waste storage. The Random Replication strategy scatters the replicas on the intermediate nodes with a ratio in random order on the way for each requested data [27]. The ratio of the replicas is determined in advance. In contrast to Path Replication, Path Random Replication is more complex to implement.

4.4 Path Adaptive Replication

The Path Replication and Path Random Replication may still cause much imbalance load as data is stored without checking the resource status. In the Path Adaptive Replication strategy, the replicas are created in the data transmission path depending on the predetermined replication ratio and node resource status, such as available storage capacity [27]. In this method, the drawback is that some attributes of other nodes like bandwidth are not considered for selecting nodes where replica hosted that may increase replica propagation time. It could also surge replicas by increasing the number of data requests.

There are some other replication methods, such as Proportional Replication, Square Root Replication, Dynamic Model-driven Replication, Dynamic Replication scheme, etc. The Proportional Replication method creates the replicas for each object that is proportional to its query distribution [26]. Square Root Replication makes replicas of an object is proportional to the square root of their query distribution [3]. In Dynamic Model-driven Replication, each node applies the strategy to its replication status and system state to get information if, when, and where the new replicas should be created [22]. Dynamic Replication technique generates the replicas to remote nodes based on the access of files frequently or if any request is done from a node to make replica [21].

Though all of those having their some advantages and disadvantages are chosen to provide data reliability for different systems according to their nature, those are not fit for the medical system for having its distinct features. For instance, Proportional Replication could create replicas into less performing nodes and become very large in number, Square Root Replication could be created into the node queried infrequently, Dynamic Model-driven Replication may copy extra replicas in the more event of node simultaneously, Dynamic Replication scheme does not provide suitable criteria for the selection of target node for hosting replica. Therefore, a new replication strategy, proposed here, could provide a better solution for the healthcare system without making replication globally and without negotiation for using each-others repositories among organizations over the world. The new strategy only needs permission to share and access data.

5. PROPOSED ARCHITECTURE

In the case of patient records, we have found some specific problems in exiting presentations. To overcome these problems, we propose a new architecture, Localized Databases' Group-based Architecture (LDGBAR). This section presents an overview of that new architecture and a new replication strategy (Area-based Replication) designed for the healthcare system.

5.1 Localized Databases' Group-based Architecture (LDGBAR)

The proposed architecture, LDGBAR, focuses on the management of different healthcare databases distributed in different places. The goal of this architecture is to allow users remote access to health records (EMR, EHR, and PHR) logically like from a centralized system. That means that the data could be stored in different locations by different healthcare organizations, and the user could access data from remote places. Also, this architecture facilitates the patients and healthcare providers to keep data on repositories avoiding data duplication. The ID Index Server, a kind of server of the proposed architecture, restricts the generating of this data duplication. Fig. 1 shows the details of the architecture. In existing database management systems, each site keeps both a local and distributed DBMS as well as its associated global data directory. The data directory consists of the location of data and database related information. Each user request is first sent to the associated global data directory to determine whether the transaction is local or global. Instead, in our proposed architecture, every time searches a short data directory created from localized databases for a user request, which saves time. In this case, the proposed architecture is designed by considering the distinct features of healthcare data. In the absence of data in localized databases, the system checks the global data directory from the ID Index Server that keeps the ID of the registered patients globally and database related information only.

The architecture consists of some components that are introduced below.

ID Index Server (IIS)

ID Index Server is responsible for keeping an index of all IDs of the patients from EMR, EHR, and PHR servers with its address locations where the data stored. The Index of ID Index Server here acts as a global data dictionary of DDBMS. As data replication is not urgent globally for having distinct features of medical records, the Index of those Servers only consists of all registered patient IDs and databases' related information instead of data. These servers maintain a peer-to-peer replication approach so that the update of any local server ID could write to all ID Index Servers. In the proposed architecture, the ID Index Server is called by DDBMS to search index in the absence of data in the group of the localized databases. It notes that the replications of data are maintained among a group of location-based databases only to ensure data availability.

PHR ID Server

PHR ID Servers keep IDs of the patient along with their data, who control their healthcare data.

EMR ID Server

These servers also keep the patient IDs and their pharmaceutical records, clinical laboratory results, patient diagnosis, disease progression, etc. managed by the specialists of a single healthcare organization.

EHR ID Server

These servers also contain the patient IDs and their healthcare data from different organizations managed by the provider.

User

Patients, Physicians, Clinicians, Professionals, Providers, Insurers, Nurses, etc. are the general users of the system.

According to Fig. 1, the user makes a direct connection to a healthcare data server that is also connected with an ID Index Server, which is shown by a solid thin line. The network of replication made locally is indicated by a solid medium line with a dotted circle. The localized nodes (databases' group from an area or city) created their replicas in each other, keep a Local ID Index. For each user request, the system first searches that Local ID Index for data. In the absence of data in localized databases, the system searches data from global databases through IIS. Besides, a solid bold line presents the network of replication made among ID Index Servers that contains the index of the patient IDs and their location addresses only without data. That network just forwards the user request made globally to the reference destination that is shown with a dotted line.

It notes that an ID is automatically added with its location address to the ID Index Server list when an ID is created in a database of the healthcare system. The change is updated to all ID Index Server by the system. When a user looks for its information, the system first searches the Local ID Index of localized servers for doing desire operation and updates the change to the replicated databases. In the absence of data in the localized servers, it connects to the IIS to get the ID of the intended patient and its reference locations where data stored of that specific ID. After getting the reference location, the connection of the ID Index Server will be released and connected to the referenced servers to perform an operation where data is stored of that patient.

5.2 Area-based Replication (ABR) Strategy

Data replica management is a pressing issue to ensure data reliability as the volume of data is increasing hugely in the medical sector. Our proposed Area-based replication strategy creates replicas on neighbor nodes surrounding the source node of data depending on the replication factor. The replication factor is an integer that indicates how many replicas of data will be generated based on the volume of data. Many remote areas where Internet speed is very slow and after performing any local access, consistency management among far nodes (or have more traffics between source and destination) would be failed repeatedly. So, Area-based replication would be more feasible for healthcare data for performing most accesses from local areas.

In the new replication technique, the system makes replicas in area basis group of databases by reducing the number of replicas. As this systems would contains small number replicas and have few traffic among nodes, so the memory consumption and the network overhead as the number of replicas is small comparatively [14]. The network overhead would also be reduced to synchronize updated data due to having a few traffic among nodes containing replicas. Also, our proposed architecture would ensure low latency according to the CAP theorem due to managing consistency only among local nodes [11].

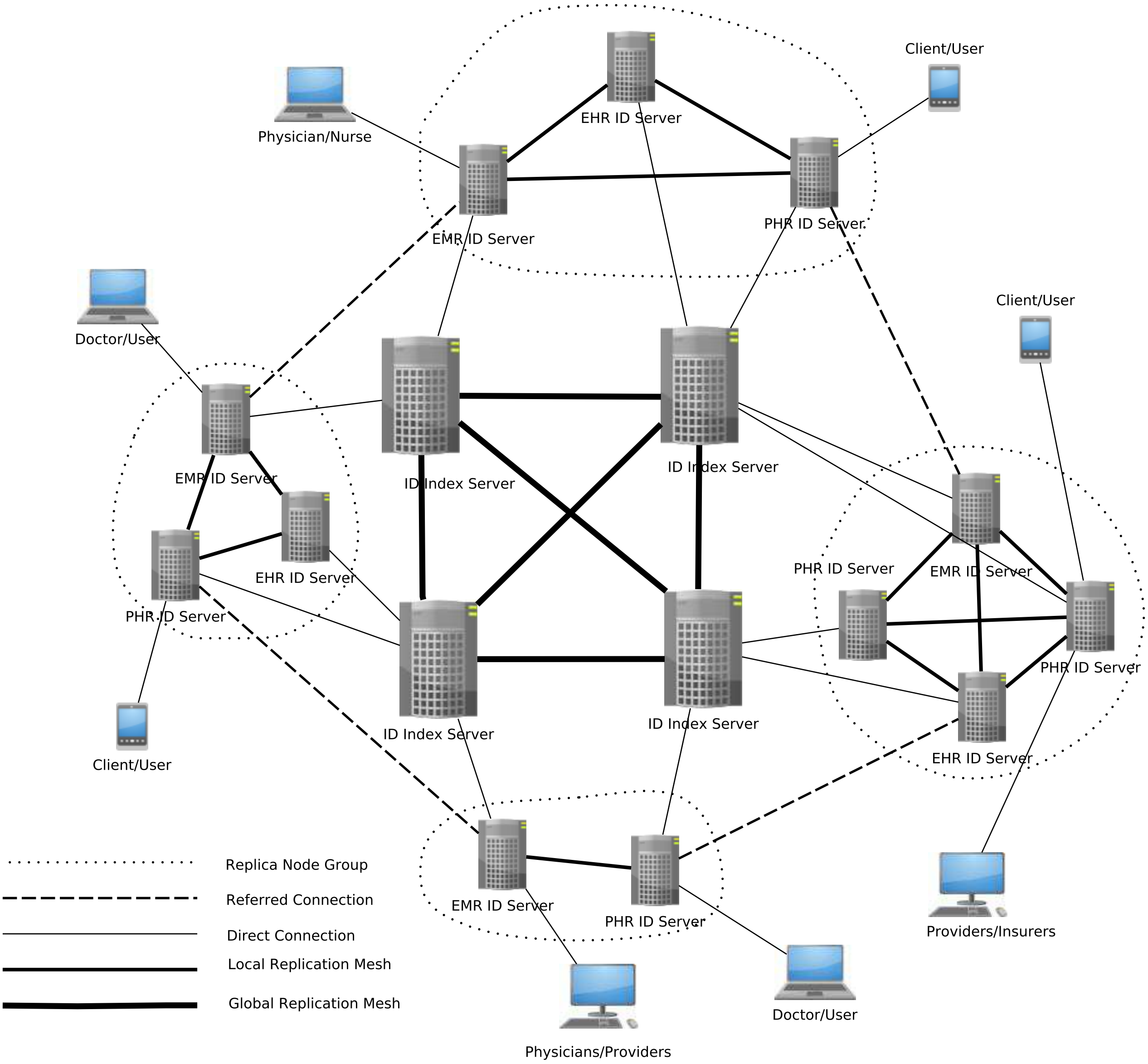

Fig. 1. Management approach of different healthcare databases (EMR, EHR and PHR) as area basis

Moreover, our proposed architecture keeps an ID Index Server for searching data quickly. Therefore, Area-based replication does not limit the performance of global access. For instance, a patient from Bangladesh may go to India for treatment, and the Indian doctor could access the previous data of that patient. According to the existing replication approaches, if this operation makes replicas on the way from Bangladesh's source node to Indian's destination node, it will consume a huge amount of storage and ultimately increase consistency maintenance overhead. The system would not here need to make replicas in those nodes because the patient back in their home country and the replica will not frequently access from there in later. So creating replication on global nodes limit the performance instead of increasing it.

Besides, this new replication strategy that limits the creation of the replicas among localized databases gives system transparency to the local organizations to run their system without suffering the hazard globally. As the system makes replicas in an area basis group of databases and contains a small number of replicas, the memory consumption would be small [14]. The network overhead would also be reduced to synchronize updated data due to having a few traffic among local nodes containing replicas. Also, the proposed architecture would ensure low latency according to the CAP theorem for managing consistency among local nodes only [11].
As a result, our proposed ABR outperforms the existing strategies to reduce concurrency complexity, network overhead, memory con-

sumption, and uses an ID index server for accessing global data to decrease the searching time.

6. METHODOLOGY

In this study, we conducted our experiment using NS3 Simulator over 250 nodes to analyze and compare data as per our strategy. The purpose of the experiment was to find out concurrency control management time, network overhead, and memory consumption of various replication strategies as well as our replication method. We here made 15 accesses for each replication technique from different nodes placed in different locations. The nodes were designed using random delay for routing the data. So, we repeated the same step 10 times for each access to get the average data. Also, we took 15 accesses with a varied number of nodes as per the rules of different replication methods. The 70% replicas of the Path Replication technique was created for the Path Random Replication, whereas 20% less for Path Adaptive Replication from Path Random Replication. To analyze data, traces and packet capture files were created from every access using the EnablePcap() method and CreateFileStream() method of AsciiTraceHelper class. Finally, the Tracemetrics tool and Wireshark analyzer were used for analyzing data from the above files. In the time of analysis, the three issues (concurrency control management time, network overhead, and memory consumption) are taken into consideration. The analyzed data are listed to compare those issues among various replication strategies as well as the proposed method.

7. DATA ANALYSIS AND DISCUSSION

The data received from the experiment are analyzed and compared among different existing replication strategies as well as our proposed methods. We show the comparison results below performed by considering concurrency management time, network overhead, and memory consumption.

7.1 Comparison of Concurrency Control Time

We compare the concurrency control time among different existing replication strategies to demonstrate how much the proposed replica method provides better performance than existing ones. In Fig. 2, we can see that the gap of the delay for concurrency management of all strategies is small at the starting of the accesses. The delay of Path Replication increases sharply with increasing the access numbers, while that of Path Random Replication and Path Adaptive Replication increases moderately. Though the rising pace for Owner Replication and Area-based Replication is small, the proposed replication strategy shows better performance. It notes that the delay of the Owner Replication will increase more with increasing the number of accesses.

7.2 Comparison of Network Overhead

Fig. 3 describes the comparison of the network overhead caused by different replica strategies. We can see from the figure that the network overhead of the various methods increases gradually with increasing the number of accesses except for our proposed replication technique. The overhead of our proposed technique indicates a stable state with an increasing number of queries. The gap in congestion between Path Random Replication and Path Adaptive Replication shows small. Remarkably, the delay of the Owner Replication that showed a smaller gap, and the Path Replication that indicated a higher gap is around 4 times and 8 times more respectively than our proposed strategy, Area-based Replication.

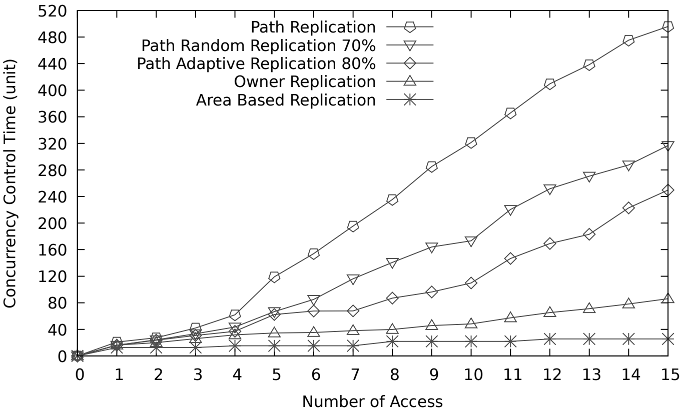

Fig. 2. Concurrency control time of different replication methods.

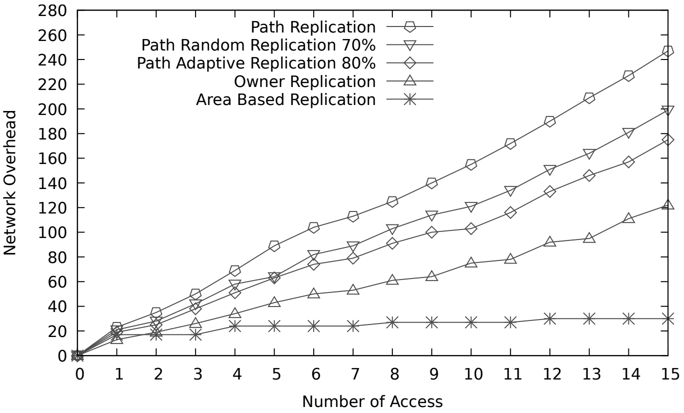

Fig. 3. Network overhead of different replication methods.

7.3 Comparison of Memory Consumption

Fig. 4 presents the memory space consumed to store replicas by the different strategies. From the figure, we can observe that the Path Replication takes the highest amount of memory, while our proposed system the lowest amount of that. It is noticeable that the memory consumption of the Owner Path Replication strategy is a bit lower, around twice of our proposed method, but that increases with increasing numbers of requests.

8. LIMITATION AND SOLUTION

The system here stores data on an area basis. So, health organizations might take extra care to back up their data for retrieving data after arising any problem in that area. One is that the health organizations keep a copy of data on another repository like a cloud outside the localized area, but no transaction is performed there by the system. Just the updated data would be stored there after a certain time and restore if needed. The other is that the local organizing group could keep their data backup in their responsibility as the data are stored location-wise; this has the benefit of retrieving the system quickly without the interrupt of all global organizations when the system unfortunately breaks. Therefore, The organizations that are agreed to ensure their data availability could keep

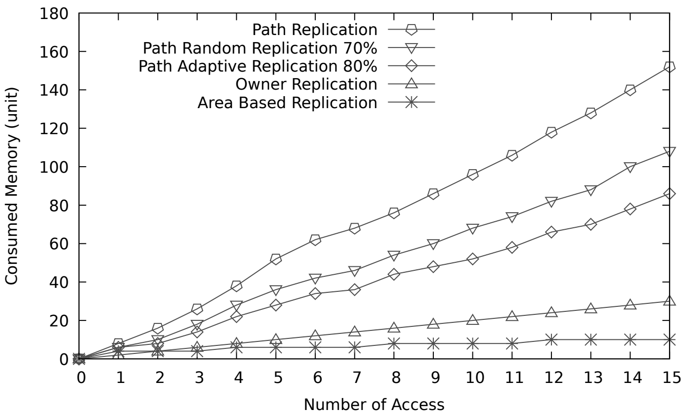

Fig. 4. Required memory of different replication methods.

data by their management, which actually could not affect the system whether other organizations do.

9. CONCLUSION

In the article, we propose an architecture, Localized Databases Group-based Architecture (LDGBAR), to share data by interconnecting the nodes of health organizations. This architecture could open a way to manage the nodes of the patient databases (EHR, EMR, PHR) scattered over the world. Users can access data swiftly from anywhere in the world after registering them under an organization. The data of EHR, EMR, PHR seem to be centralized from the viewpoint of healthcare users logically, but those databases are physically decentralized. That means that the proposed architecture ensures data availability from everywhere through managing the databases distributed over the world by mitigating many problems and barriers. The architecture could benefit from various aspects: any organization could come under this healthcare system and connect without any global negotiation; avoid the creation of duplicate ID, etc.

We also present an Area-based Replication (ABR) strategy fitting the nature of the patient data for reducing the numbers of replicas and traffic among them. The performance evaluation of the proposed replication method compared to existing replication methods exposes several remarks. Performing the equal number of accesses, the concurrency management time of the ABR technique outperforms the other existing replication methods. The network overhead of the proposed technique shows around 25% of the smallest network overhead of the existing ones. The memory consumption of the ABR procedure is also reduced significantly than other methods. Besides, as we keep the replicas among localized databases considering the distinct features of healthcare data, the network overhead shows smaller than other traditional strategies. Moreover, for storing replicas on nearer nodes and having control of data on a local wised group, the system could be retrieved quickly in case of crashes. The ABR also benefits the localized organizations' system transparency to run the system without suffering hazard globally.

Therefore, the proposed architecture and Area-based Replication would be more feasible for healthcare data to manage distributed databases and share data over the world by mitigating different challenging issues.

Acknowledgment

The authors would like to thank to the Information & Communication Technology Division, Bangladesh for their financial support.

10. REFERENCES

[1] WSW Awang, MM Deris, Omer F Rana, M Zarina, and ANM Rose. Affinity replica selection in distributed systems. In *International Conference on Parallel Computing Technologies*, pages 385–399. Springer, 2019.

[2] Kyungyong Chung and Roy C Park. P2p-based open health cloud for medicine management. *Peer-to-Peer Networking and Applications*, pages 1–13, 2019.

[3] Vassilios V Dimakopoulos, Spiridoula Margariti, Mirto Ntetsika, and Evaggelia Pitoura. Data replication in p2p systems. In *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications*, pages 589–615. IGI Global, 2010.

[4] Sue S Feldman, Scott Buchalter, and Leslie W Hayes. Health information technology in healthcare quality and patient safety: literature review. *JMIR medical informatics*, 6(2):e10264, 2018.

[5] Xiangzhu Gao and Jun Xu. Towards an e-health ecosystem for china. In *International Conference on Smart Health*, pages 48–60. Springer, 2019.

[6] Munish Gupta and Heather C Kaplan. Improving quality improvement in neonatal-perinatal care. *Clinics in perinatology*, 44(3):xvii, 2017.

[7] M Harika, S Rahmadika, and DR Ramdania. Blockchain technology for managing an architectural model of decentralized medical record. In *Journal of Physics: Conference Series*, volume 1402, page 077027. IOP Publishing, 2019.

[8] Yahya Hassanzadeh-Nazarabadi, Alptekin Küpçü, and Öznur Özkasap. Decentralized and locality aware replication method for dht-based p2p storage systems. *Future Generation Computer Systems*, 84:32–46, 2018.

[9] Yahya Hassanzadeh-Nazarabadi, Alptekin Küpçü, and Oznur Ozkasap. Decentralized utility-and locality-aware replication for heterogeneous dht-based p2p cloud storage systems. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

[10] Yahya Hassanzadeh-Nazarabadi, Alptekin Küpçü, and Öznur Özkasap. Lightchain: A dht-based blockchain for resource constrained environments. *arXiv preprint arXiv:1904.00375*, 2019.

[11] Martin Kleppmann. A critique of the cap theorem. *arXiv preprint arXiv:1509.05393*, 2015.

[12] Rani Kumari, Parma Nand, and Rani Astya. Integration of blockchain in wban. In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 144–149. IEEE, 2019.

[13] Elias Leontiadis, Vassilios V Dimakopoulos, and Evaggelia Pitoura. Creating and maintaining replicas in unstructured peer-to-peer systems. In *European Conference on Parallel Processing*, pages 1015–1025. Springer, 2006.

[14] Chunlin Li, YiHan Zhang, and Youlong Luo. Adaptive replica creation and selection strategies for latency-aware application in collaborative edge-cloud system. *The Computer Journal*, 2019.

[15] Jason Lucas. Systems and methods for asynchronous distributed database management, February 7 2013. US Patent App. 13/564,242.

[16] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, pages 84–95, 2002.

[17] Gunasekaran Manogaran, Chandu Thota, Daphne Lopez, V Vijayakumar, Kaja M Abbas, and Revathi Sundarsekar. Big data knowledge system in healthcare. In *Internet of things and big data technologies for next generation healthcare*, pages 133–157. Springer, 2017.

[18] Joao Paiva, Joao Leitao, and Luis Rodrigues. Rollerchain: A dht for efficient replication. In *2013 IEEE 12th International Symposium on Network Computing and Applications*, pages 17–24. IEEE, 2013.

[19] Bambang Eka Purnama. Distributed data patient in medical record information system. *International Journal of Scientific & Technology Research*, 2020.

[20] Xiaogang Qi, Min Qiang, and Lifang Liu. A balanced strategy to improve data invulnerability in structured p2p system. *Peer-to-Peer Networking and Applications*, 13(1):368–387, 2020.

[21] Sathish Rajasekhar, Bin Rong, Kwong Yuen Lai, Ibrahim Khalil, and Zahir Tari. Load sharing in peer-to-peer networks using dynamic replication. In *20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)*, volume 1, pages 1011–1016. IEEE, 2006.

[22] Kavitha Ranganathan, Adriana Iamnitchi, and Ian Foster. Improving data availability through dynamic model-driven replication in large peer-to-peer communities. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02)*, pages 376–376. IEEE, 2002.

[23] Alex Roehrs, Cristiano André da Costa, and Rodrigo da Rosa Righi. Omniphr: A distributed architecture model to integrate personal health records. *Journal of biomedical informatics*, 71:70–81, 2017.

[24] Bingqing Shen, Jingzhi Guo, and Yilong Yang. Medchain: efficient healthcare data sharing via blockchain. *Applied sciences*, 9(6):1207, 2019.

[25] Haiying Shen. Irm: Integrated file replication and consistency maintenance in p2p systems. *IEEE transactions on Parallel and Distributed systems*, 21(1):100–113, 2009.

[26] Saurabh Tewari and Leonard Kleinrock. Proportional replication in peer-to-peer networks. In *INFOCOM*, pages 1–12, 2006.

[27] Hiroshi Yamamoto, Daisuke Maruta, and Yuji Oie. Replication methods for load balancing on distributed storages in p2p networks. *IEICE transactions on information and systems*, 89(1):171–180, 2006.