ISSN:0975-9646
Anuraag Vikram Kate, et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 8297-8299

Sector Based Cloud Storage of Blood Groups

[1]Anuraag Vikram Kate, [2] Nikilav P V, [3]Praneeth Ramesh, [4] Sriniketh R, [5] Naren J

[1, 2, 3, 4] B.Tech Student CSE, School of Computing, SASTRA UNIVERSITY, India,
[5] Assistant Professor CSE, School of Computing, SASTRA UNIVERSITY, India,

***Abstract-* In this ever perilous world, with accident rates rising, the necessity for fast and proficient assistance from blood banks, to save human lives, is increasing. The prevailing system for request of blood by hospitals is inept and sluggish and thus leads to loss of life. In this paper, we propose and explore a framework that exhibits the use of cloud computing to store sector based blood group data for a state or even a country. This allows the hospitals to omit several counter-productive steps to acquire blood for a patient in need. The cloud-based structure will contain the various blood groups available at each of the blood banks, present in each pre-defined sector of the state or country. It facilitates the hospitals to retrieve data rapidly and contact the closest blood bank that has the blood group that is necessary.**

***Keywords-* Medical Sciences, Cloud Computing, Information Storage Framework**.

I. INTRODUCTION

Sector Based Cloud Storage of blood groups is a sequential, methodical and swift process of accessing the blood bank data from a cloud server. The prevailing framework of utilizing a mobile application to check for blood donors based on a cloud server is laborious because, allowing people to access donor information, apart from being a big security risk, is not proficient enough. The model we propose, allows hospitals to access blood bank information, which is stored in a cloud server, to purely check if the necessary blood group is available in any blood bank in that particular sector. This substantially speeds up the procedure of blood retrieval, as the bank can easily skip through various rounds of calling each blood bank manually and checking if that blood group is obtainable. Through this system, hospitals all around the country can get the particulars of the closest blood bank with the required blood group and thus increase the chances of saving the patient.

II. PROBLEM ASSESSMENT

The current system of blood retrieval used by hospitals is so drawn out, that even the trivial process of determining whether a specific blood group is available at the blood bank or not, has several superfluous protocols. This results in numerous delays which in turn puts several patients' lives at risk. We propose a system to eliminate this delay and speed up the process of blood retrieval [1]. There is an android application for blood donors, which enables people to fix appointments to donate blood. It also allows access to blood levels in blood banks. These details are updated at the end of every working day. Here the donors are required to send in their details and personal information. Also this application can be used to request donors who are registered to come forward and donate. But this makes the retrieval of blood very slow and tedious.

III. CONTRIBUTION

In this paper, we put forth a system to assist in blood retrieval by allowing the hospitals to check if the blood banks in their sector have the blood group that they require without having to go through the various protocols. If a blood bank has the required blood group, its details are displayed and if there are more than one, then they will be ordered according to which one is geographically closer to the hospital. If, however, no blood bank in the sector has the required blood group, then the hospital can check in the next sector. All the data from the blood banks is stored in a cloud server which can be accessed by any hospital in the country.

IV. RELATED WORK

A. Virtual Blood Bank Project

The Virtual Blood Bank project is based in New Delhi. The system is implemented using java and web applications. This system allows hospitals to find donors using their residential address [2].

B. Emergency Blood Bank directories using www.bloodbanker.com

This website holds the details of blood banks and hospitals in each state in the USA. The website can be used to locate the nearest blood bank or hospital. It is a widely used service and was the first of its kind.

C. Location based Blood Bank using cloud storage

This proposed model is based on a mobile app which is linked to a cloud server. Donors can register themselves on this app and their details are stored in the cloud storage. These details can be retrieved by anyone with the app on their phone to locate the donors in case of an emergency [3].

V. FRAMEWORK

A. Website and Request Generation

This process of searching for a specific blood group is initiated by the hospital using a website. This website consists of an uncomplicated GUI which requests the bank for the following details.

- Blood Group
- Quantity of Blood required
- Location of Hospital

Once these details are provided, the website accesses its cloud storage where all the blood bank data is stored [4].

B. Location detection

Using the hospital's location, which is obtained through the GUI, the nearest blood banks are detected using GPS. This will enable the representation of results based on geographic proximity. This also facilitates the process of locating the closest possible blood bank that has the required blood group and quantity.

C. Displaying Results

The given blood group and quantity is searched for in the cloud database, where the blood bank data has been stored. When the results are found, they are displayed on the website for the hospital to see. The results contain the basic information of the blood banks that have that specific blood group, ordered by the geographical proximity.

D. Architectural Design

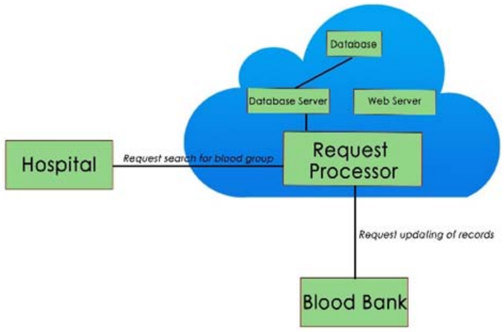

Fig. 1. Architectural Design of the system

This framework portrays and depicts the overall structure and functioning of the model proposed. The hospital accesses the website whenever there is a need for blood groups. A query is sent by the hospitals via the website to the request processor that accesses its database and sends the availability of the groups back to the hospitals. The availability of groups is ordered by geographical proximity. This is determined by the location of the hospital and the location of the blood banks. On the other end of this system, the database is populated and updated by various blood banks at regular intervals, say at the end of every day [5].

VI. IMPLEMENTATION

The web app is implemented on the cloud. We have used the Google App Engine cloud platform for this purpose [6]. It consists of two interfaces – one for the blood bank and the other for the requiring hospital. The interface for the blood bank consists of several fields which enables the bank for updating the blood data – the blood group and the amount of blood available in each group. This data is stored in the cloud and is updated on a daily basis. The hospitals are presented with an alternate interface which provides them with fields for requesting blood. When the request is sent from the hospital, the database in the cloud is accessed and the process of retrieval of information commences. The simulation of retrieval of data has been implemented with the help of the Google Maps JavaScript API V3 and the Google Places API [7]. The vicinity radius is 5 Km: the application catering to the hospital acts as the epicentre and searches for all the blood banks within the specified radius [8]. Using the location of the blood bank and the location of the hospital, the blood bank nearest to the hospital is considered first. Once the blood bank has been singled out, the application focuses on the aspect of retrieving the needed blood group and the amount specified from the database stored in the cloud [9]. If the data which includes the blood group and the amount are available in the cloud, the hospital immediately contacts the blood bank with the available resource for procuring the blood. In the case when the required amount or the blood group is not available in the particular blood bank, the next nearest blood bank is chosen and the process is repeated. This is done by using the Google Places API which has an optional parameter 'rank by distance': it orders the blood banks based on the distance between the requesting hospital and the blood banks. As a result of this, the banks which are nearer to the location of request are searched sector-wise and a notification is sent to the hospitals based on the obtainability [10].

VII. SECURITY ENHANCEMENTS

The current system where the Android application requests for blood donors whenever there is the need of it has imminent security threats. This not only endangers the personal details of the donor but also gives way for malicious users to obtain confidential data for their use. The cloud system reduces this threat significantly as the data is stored in a secure cloud. It also avoids the risks involved in the functioning of a broadcast network. An additional layer of security has been employed by encrypting the requests from the hospital and the data inputted by the blood banks in the cloud [9]. The algorithm for encryption and decryption using RSA is as follows.

A. RSA Algorithm

1) Calculation of keys:

1. Randomly select two huge random prime integers: a &b.
2. Calculate x and φ(x): x = ab and φ(x) = (a-1) (b-1)
3. Select an integer f, 1 < f < φ(x) where: gcd (f, φ(x)) =1.
4. Calculate z, 1 < z < φ(x) such that: fz ≡ 1 (mod φ(x)).
5. The public key is (x, f) and the private key is (x, z).
6. The values of a, b and φ(x) are private.
7. f is the public exponent.
8. z is the private exponent.

2) Encryption of plain text:

The cipher text C is computed by the formula '$C = M^f \bmod x$' where M is the plain text.

3). Decryption of cipher text:

The plain text M can be calculated from the cipher text C by the formula '$M = C^z \bmod x$'.

This method of ciphering the requested data enables for a more secure environment [10]. The cloud system also increases the efficiency and speed as the hospitals do not have to waste time contacting individual blood banks for blood, which ends up risking the patient's life. In the existing system, there is no preliminary contact between the hospitals and the blood banks and that is a major concern due to the innumerable loss of lives. The proposed system, however, is much more efficient as the contact to the blood bank is made only after the hospital receives the details of the blood bank containing the required amount and blood-type, thus saving ample time [13].

VIII. FUTURE IMPLEMENTATION

The current scope of the implementation ends with improving the sluggish and phlegmatic system for blood retrieval. But the system can be extended so that the time taken to deliver the blood to the hospital is also reduced by automating the contact and retrieval process as well.

In the proposed system, the blood banks upload the details pertaining to the blood group and the amount of blood they possess on a daily basis. But this method has a drawback: the data updated might be outdated and the blood banks might not have the blood though it is indicated as available in the database. By using historical data, a more efficient system can be put in place. Machine learning enables us to foresee the amount and the groups of blood the blood banks will have the next day by using the data obtained from the past through the process of supervised learning.

IX. CONCLUSION

Thus based on the findings, we conclude that the amendments presented by our framework clearly enrich the feasibility and the efficacy of the existing blood blank retrieval system. Again, we focus only on the retrieval of availability of blood data in nearby blood banks rather than the delivery of the blood to the hospitals. This proposed framework clearly depicts the shortcomings of the existing framework and promises to improve its adeptness.

REFERENCES

[1] Borko, Furht,”*Handbook of Cloud Computing*”, First Edition, Dec 2010, Springer.

[2] Gawali Akshay, Mayuresh Bhyote, Anamika Lokhande, Poorva Mehendale: Location Based Blood Bank using Cloud Storage, IJRME, Volume 3 Issue 3

[3] Singh Ramesh, Preeti Bhargava, and Samta Kain: Smart Phones to the Rescue: The Virtual Blood Bank Project, *2007*.

[4] Arlitt MF, CL Willaimson: Internet Web Servers, IEEE/ACM Transactions on Networking, 1997

[5] Stonebraker M, R Agrawal, U Dayal: DBMS Research at a Crossroads: The Vienna Update, VLDB, 1993

[6] Vouk M A: Cloud Computing – Issues, Research and Implementations, Journal of Computing and Information Technology, 2004

[7] Maurizio Gibin, Alex Singleton, Richard Milton, Pablo Mateos, Paul Longley. 2008. An exploratory cartographic visualization of London through the Google Maps API. *Springer: Applied spatial analysis and policy, Volume 1 Issue 2* (July.. 2008), 85-97.

[8] Eugene Ciurana. Developing with Google App Engine.

[9] Dan Sanderson. Programming Google App Engine: Build & Run Scalable Web Applications on Google’s Infrastructure. (October.. 2012)

[10] Zhang Qi, Lu Cheng, Raouf Boutaba: Journal of Internet Services and Applications May 2010, Volume 1, Issue 1

[11] Feng Deng-Guo, Zhang Min, Zhang Yan, Xu Zhen. Study on cloud computing security. *CNKI.*

[12] Xin Zhou, Xiaofei Tang. 2011. Research and implementation of RSA algorithm for encryption and decryption. *IEEE Xplore* (Aug.. 2011), 1118-1121.

[13] Michael Armbrust, Armando Fox, Rean Griffith, Anthony Joseph, D., Randy Katz, Andy Kowinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia. A view of cloud computing. *Communications of the ACM* (April.. 2010), 50-58. DOI=10.1145/1721654.1721672