Journal of Scientific & Industrial Research
Vol. 79, December 2020, pp. 1106-1109

Performance Evaluation of the Cloud-based QR Code Identity Tag System with Cloudlets

Vassilya Uzun
Department of Computer Engineering, Istanbul Gedik University Istanbul, Turkey

Received 26 April 2019; revised 26 October 2020; accepted 03 November 2020

In this paper, a QR Code Identity Tag System designed for Turkish healthcare and served through the cloud is presented. The system designed is a distributed information management system as the medical data objects are geographically distributed over the system. The data objects are geographically distributed as they are placed on servers closest to the geographical location where they are most frequently required. This enables that the data is always available and the access is provided in minimum time. Additionally, to improve the performance the system employs Mobile Edge Computing technology in the form of cloudlets. The simulation results show that system performance improves as the number of cloudlets used increases and popularity threshold decreases.

Keywords: Cloud computing, E-health, Mobile Edge Computing, Object retrieval, Personal Health Records

Introduction

This paper proposes a redundant hierarchical approach to place the personal health record (PHR), electronic health records (EHR), electronic medical records (EMR), along with all necessary clinical data in a cloud-based server system, where a cloud and mobile edge computing (MEC) cloudlet technology is employed to provide the infrastructural and data center facilities necessary for storing and managing all medical objects belonging to the QR Code Identity Tag System.[1]

The main benefit of using clouds in healthcare is provision of instant access to the service users, while cloudlets enable improved privacy and reduced latency, bandwidth, scalability, reliability, and energy efficiency.[2–14]

Materials and Methods

Distributed cloud-based scheme presented in this paper uses a layered approach towards distributing and placing object in the system. A central cloud holding every data object pertaining to the system is used along with a set of servers that serve each major geographical unit (district) of the country to which the system belongs. Hierarchically speaking, these servers are arranged in layers where each sub-district is below the district level servers. These district level servers are below the central server in the layered approach. Every lower layer requests the necessary records from the layer above as and when needed.

The object placement works in the following manner: each object, from a text report to the video records of procedures, is initially paced in the central cloud system. Whenever a request is received from a client, the system takes the request and provides it from the central cloud through the requesting node to the client and increments its request count. Whenever a request count for a particular record passes a certain threshold, the requested object is copied to the server below the current server holding the object, a step nearer to the requesting node, until it finally and eventually reaches the network edge, provided enough requests for the same record have been made. This way records used in a certain geographical location get aggregated in servers serving those geographies. Similarly, least recently used (LRU) data objects are removed from the lower layers although the original copy of the record on the central cloud is never removed/ deleted.

Physical communication connections exist between the servers of different layers. All connections are duplex and hence can be used for bidirectional communication. The network architecture used is based on Red-HI. In Red-HI a node is connected to exactly two nodes of the higher layer in the hierarchy; the redundancy in Red-HI is that of links and not bandwidth. This means that instead of doubling the bandwidth to increase the link, the available

Author for Correspondence
E-mail: vassilya.uzun@gedik.edu.tr

bandwidth is divided among the two links. This gives a higher degree of connectivity and it is a solution to the bottleneck problem. This also provides continued service provision in face of node/link failures.

The cloudlet is placed nearby the hospital/ region and covers an area that can be accessed by authorized users who can access the information of patients and remotely follow their status.

The mobile traffic can be reduced from one to two thirds by cashing the content at the core of the mobile network. Popularity of the content can be used as a parameter that helps the system decide on what should be cashed at the edges of the network. Increasing the probability of availability of content requested at the edges, aids in maximizing the hit probability of the content at the cache.[9–12] Therefore, in our architecture, the cloudlets follow the same popularity scheme as the servers on the cloud.

The system proposed is scalable in that any number of layers of servers can be added with minimum modifications to the hardware. It can serve a system of a few clinics, to a national health ministry consisting of many large-scale hospital networks. The system depicted in Fig. 1 is one where a dual layer system is used.

Results and Discussion

This study investigates the performance of the proposed system through simulations, developed using the GPSS World simulation tool.[15] The values of the simulation parameters are given in Table 1.

The propagation delay of links is assumed to follow a normal distribution. A mean of 0.07 s with SD of 0.014 is assumed. With a rate as λ, the flow of all the requests is modeled as a Poisson process. The arrival rates of 0.05, 01, 0.15, and 0.2 are considered for obtaining the simulation results.

The service rates for the cloudlets are set to 15.

Our simulation model consists of an entry level layer with 5 servers, intermediate level layer 1 with 4 servers, intermediate level layer 2 with 3 servers, and a root level layer with 2 servers. This means that our model consists of 4 cities, each city is divided into 5 districts. Moreover, the country is divided into 3 regions and the 2 main servers are located in the two parts of the country.

There are 3 hospitals in each city district, so the server located in each region is responsible for the local hospitals and the cloudlets.

The proposed system is evaluated under the 6 scenarios:

Scenario 1: no cloudlets are allocated, popularity threshold = 25.

Scenario 2: one cloudlet is allocated to each region, popularity threshold = 25.

Scenario 3: one cloudlet is allocated to each hospital, popularity threshold = 25.

Scenario 4: no cloudlets are allocated, popularity threshold = 50.

Scenario 5: one cloudlet is allocated to each region, popularity threshold = 50.

Scenario 6: one cloudlet is allocated to each hospital, popularity threshold = 50.

Performance Measures

The evaluation of the system was performed in terms of the following performance measures: *Load*,

Table 1 — Simulation parameters and their corresponding values

Request arrival rate	0.05, 0.1, 0.15, 0.2 s^{-1}
Popularity threshold	25, 50
Storage size of an object	0.5 GB
Bandwidth requirement of an object	1 Mb/s
Total number of objects	200

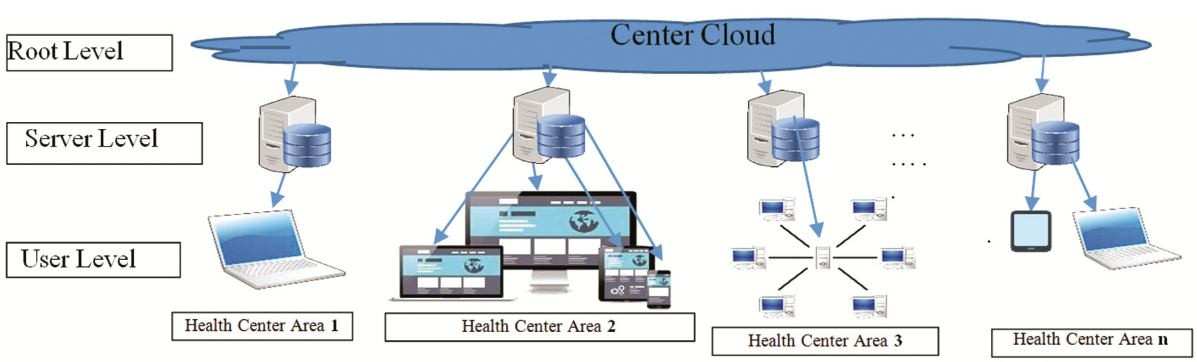

Fig. 1 — Cloud architecture

Average Transmission Delay (*ATD*) and *Blocking Ratio* (*BR*).

ATD is defined as follows:

$$ATD = D_s / N_s \quad \ldots (1)$$

where D_s is the total delay of successful requests and N_s is the total number of successful requests.

BR is defined as follows:

$$BR = N_b / N_t \quad \ldots (2)$$

where N_b is the total number of blocked (rejected) requests and N_t is the total number of requests.

Load of the servers is provided by the simulation tool.

Simulation Results

The simulation results of the system under the 6 scenarios are shown in Fig. 2, 3 and 4.

As could be seen in Figs. 2–4, the developed model running under the *Scenario 3* significantly outperforms other 5 scenarios in terms of *Load*, *ATD* and *BR*. This means that increasing the number of cloudlets positively effects the performance of the system, which is due to the caching at the cloudlets.

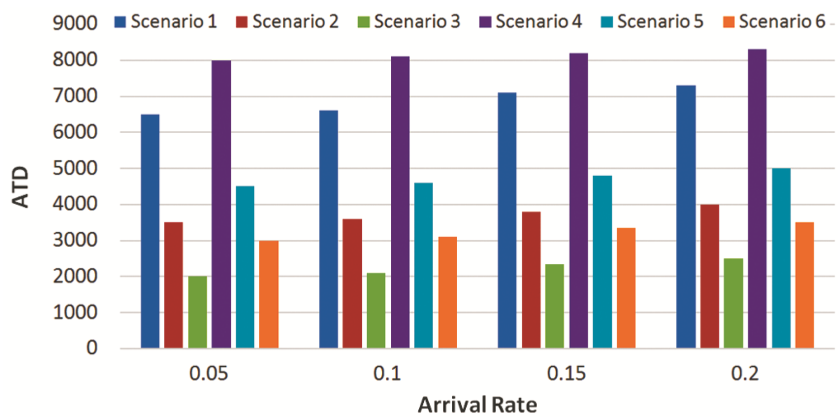

Fig. 2 — ATD versus arrival rate

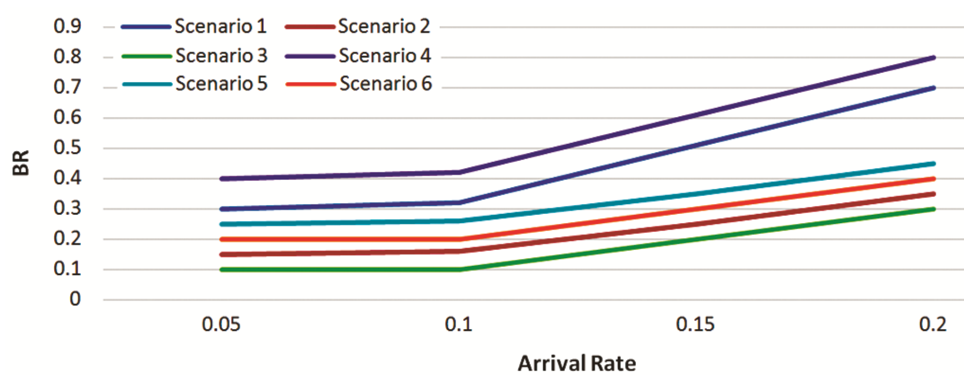

Fig. 3 — BR versus arrival rate

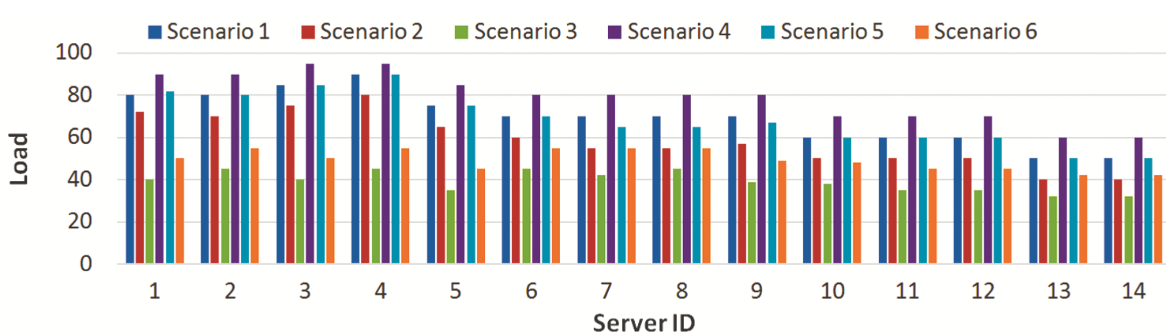

Fig. 4 — Server ID versus Load with arrival rate of 0.1

On the other hand, price of the system will rise as number of cloudlets rises. Therefore, the decision on the number of cloudlets to be used should be made based on the financial resources available.

Moreover, the system performs better with popularity threshold of 25. The performance is better because popular data objects, that are objects that reach a threshold of requested number of times, are placed in location nearest to their destination, thus reducing the transmission delay and the blocking ratio.

To sum up, the best performance is achieved when one cloudlet is assigned to each hospital and the popularity threshold is assigned the value of 25.

Conclusions

An object placement technique is presented that uses a popularity threshold to place the objects in the distributed cloud-based system. The popularity threshold of a particular object is updated every time a request is made for that object. As the storage constraints get stringent, Least Recently Used objects marking the least popular objects are removed from the densely populated servers/nodes. The removed objects are always available in the central cloud. Moreover, cloudlets were used in this architecture to reduce the load on the cloud and to reduce the latency.

This study showed through simulations that employing cloudlets and popularity threshold improves the quality of service (QoS) of a networked healthcare system by reducing the delay, reducing load on the cloud, improving the connectivity, and fault tolerance of the network.

References

1 Uzun V & Bilgin S, Evaluation and implementation of QR Code Identity Tag system for healthcare in Turkey, *Springerplus*, **5(1)** (2016) 1454.

2 Dewangan B K, Agarwal A, Choudhury T, Pasricha A & Chandra Satapathy S, Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges, *Softw: Pract Exper*, (2020) 1–20.

3 Sharma T, Choudhury T & Kumar P, Health Monitoring & Management using IoT devices in a cloud based framework, in *International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Jun 22–23, 2018.

4 Khanna A, Sah A & Choudhury T, Intelligent Mobile Edge Computing: A deep learning based approach, in *International Conference on Advances in Computing and Data Sciences*, 24–25 April 2020.

5 Kumar A Nirmal, Efficient utilization of virtual instances by suspend resume strategy in cloud data center, *J Sci Ind Res*, **79(06)** (2020) 526–530.

6 Yang Y, Li X, Qamar N, Liu P, Ke W, Shen B & Liu Z, Medshare: a novel hybrid cloud for medical resource sharing among autonomous healthcare providers, *IEEE Access*, **6** (2018) 46949–46961.

7 Muhammed T, Mehmood R, Albeshri A & Katib I, UbeHealth: a personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities, *IEEE Access*, **6** (2018) 32258–32285.

8 Kavitha R, Kannan N, Nazneen R & Zubar H A, Cloud computing integrated with testing to ensure quality, *J Sci Ind Res*, **75** (2016) 77–81.

9 Wang S, Zhang X, Zhang Y, Wang L, Yang J & Wang W, A survey on mobile edge networks: Convergence of computing, caching and communications, *IEEE Access*, **5** (2017) 6757–6779.

10 Abbas N, Zhang Y, Taherkordi A & Skeie T, Mobile edge computing: A survey, *IEEE Internet Things J*, **5(1)** (2018) 450–465.

11 Mach P & Becvar Z, Mobile Edge Computing: A Survey on Architecture and Computation Offloading, *IEEE Commun Surv Tutor*, **19(3)** (2017) 1628–1656.

12 Mao Y, You C, Zhang J, Huang K & LetaiefKB, A Survey on Mobile Edge Computing: The Communication Perspective, *IEEE Commun Surv Tutor*, **19(4)** (2017) 2322–2358.

13 Kumar P, Gupta A & Kumar S, Dynamic Key Based Algorithm for Security in Cloud Computing Using Soft Computing and Dynamic Fuzzy Approach, *J Sci Ind Res*, **78(09)** (2019) 596–600.

14 Sen S & Sen S, iPaaS–Cloud Solution for the Cloud Problem, *Sci Rep*, **55(12)** (2018) 44–45.

15 www.minutemansoftware.com/ (26 October 2020).