Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 3137 – 3153
© TÜBİTAK
doi:10.3906/elk-2001-156

Optimization of real-time wireless sensor based big data with deep autoencoder network: a tourism sector application with distributed computing

Bekir AKSOY[1,*], **Utku KÖSE**[2]

[1]Department of Mechatronics Engineering, Faculty of Engineering, Isparta University of Applied Sciences, Isparta, Turkey

[2]Department of Computer Engineering, Faculty of Engineering, Süleyman Demirel University, Isparta, Turkey

Received: 30.01.2020 • **Accepted/Published Online:** 05.06.2020 • **Final Version:** 30.11.2020

Abstract: Internet usage has increased rapidly with the development of information communication technologies. The increase in internet usage led to the growth of data volumes on the internet and the emergence of the big data concept. Therefore, it has become even more important to analyze the data and make it meaningful. In this study, 690 million queries and approximately 5.9 quadrillion data collected daily from different servers were recorded on the Redis servers by using real-time big data analysis method and load balance structure for a company operating in the tourism sector. Here, wireless networks were used as a triggering factor to gather data from visitors of the hotels and the analysis was supported with an optimization approach through the deep autoencoder network. According to the data density gathered from the structure developed with distributed computing and the API software in C# language, server group numbers were increased to list the most affordable hotel in the desired times. Thanks to the developed architecture and software, response times of the servers were significantly reduced. In detail, it was seen that the HAProxy responded 11 times faster than NetScaler as the new architecture responded 1160 times faster than the old one. Also, the HashSet system in the newly developed architecture responded 18 times faster than the List system and as general, the new architecture was found to be 9 times faster than the old architecture.

Key words: Big data, Redis, wireless sensors, autoencoder, distributed computing, tourism

1. Introduction

The international tourism sector has been an important source of external revenue as one of the fast growing sectors in many countries of the world [1]. Therefore, the tourism sector is an important driving force for many countries' economy [2]. In a report published by the World Tourism Organization in 2000, the increase in the number of tourists by 2020 is estimated to reach 1602 million [3]. Digital technology and social media play an important role in the rapid development of the tourism sector [4]. Tourism agencies and hotel businesses can communicate with customers quickly by using social media tools such as Facebook and Twitter [5] or internet forum sites such as sina.com, ctrip.com [6]. Tourists' impressions and comments about their holidays especially on social media sites [7–10] and websites such as Tripadvisor, Yelp [11–14] play an important role in tourists' decision-making processes. Information sharing over social media and commenting sites is found more reliable by customers [15]. So, online commenting sites have great importance for tourism entrepreneurs [16].

The increase of information sharing in digital technology has led to an increase in online reservations in the tourism sector. According to Ctrip data, the demand for online accommodation increased by 28.8% between

*Correspondence: bekiraksoy@isparta.edu.tr

July 2016 and June of 2017 [17]. That caused to the growth of the data obtained from the tourism sector. In the literature, big data is first defined as 3V (velocity, volume and variety), then as 4V (velocity, volume, variety and veracity) [18–21]. Velocity is to express the speed of received/sent data, volume is used to express the amount of data, the variety is to express data type [19] and by McKinsey company the term veracity is used to express the reliability of data [22]. The term "value" added to the definition of big data as the fifth V that refers to data value. Value is used to reduce the time and transaction complexity by determining which data set is to be included in data analytics [23]. Evaluation, which is the whole of large-scale calculations, system analysis and storage of data to extract information effectively, is possible with big data [24, 25].

Big data applications are successfully used in many areas such as banking, insurance, health, education, spatial data, and the detection of online customer behavior [26–32]. Tourism sector is one of the sectors where customer accumulations are found to be important. When the academic studies related to the use of big data in the tourism sector are examined, a model that provides a tour plan to the tourists by using the big data such as tour cost, traffic density and weather conditions is proposed [33]. In their study, Miah et al. analyzed tourist behaviors with the help of the Flickr program for the photos taken in Melbourne, Australia and shared by tourists on the social media and proposed a new method using big data analytics in decision making process [34]. In their study, Salas-Olmedo et al. analyzed the big data, which consists of the digital footprints of the tourists, by analyzing three different sources such as Panoramio (trip), Foursquare (consumption) and Twitter (connected accommodation) instead of a single source for tourists' presence in the region and demonstrated the advantages of the analysis over a single source analysis [35]. In another study, the big data analytics approach based on customer information management was used for the Apulia region of Italy to transform the data gathered from the social networks into a competitive form for tourism destinations [36]. Fuchs et al. used the big data analysis for the tourism activities such as web navigation and reservation by conducting business intelligence approach to find out the pereferred locations by tourists [37]. Sun et al. proposed an approach based on the concept of smart and connected communities (SCC) that combines IoT and big data analysis for tourism and sustainable cultural heritage for the city of Trento, Italy [38]. In the associated literature, more research works were done in terms of analyzing tourism data. Here, especially batch analysis, real-time analysis and hybrid computation are employed big data methods [39–42]. At this point, the performance of real-time data for the hotel bookings and search operations are better than other big data analysis methods [43].

Distributed computing systems are used to solve data processing problems on large amounts of data and to improve the process performance [44]. Each processor in the distributed computing system has different performance and reliability features. For using the processing power in distributed computing systems, it is necessary to optimize system performance indices such as run-time or cost of the program modules [45]. In the design of distributed computing systems, dividing task by using allocation and load balancing has great importance. Thus, in distributed computing system, the application can be divided into several tasks and run in dissimilar nodes [46]. The performance of the applying in the distributed computing system depends on the deployment of the application tasks to the nodes, which is called task allocation problem [47, 48]. If more tasks are broken into specific nodes, tasks are transferred from heavily-loaded nodes to light-load tasks by using the load balancing logic to lessen the waiting time of tasks on nodes [49]. It is suitable to use the task queues collection and a simple load balancing system for scheduling shared memory tasks of parallel machines. Task planning on such machines is usually carried out through a single, globally accessible work environment [50]. When the literature on distributed computing is examined, it is seen that smart computing route suggestion system for tourists was developed by using cloud computing and distributed computing to ensure transaction security and time stability of the stored data [51]. Jiang et al. made a cloud computing smartphone app for tourists wanting to homestay in Taiwan [52]. Load balancing allows ensuring the performance of a parallel and distributed system by means of redistribution of load between processors [53, 54].

Load balancing means an even distribution of workload on all nodes. Load balancing gives accurate results to the user by using the resources correctly. Therefore, an appropriate load balancing algorithm must be used when selecting virtual machines or servers [55]. Load balancing is a major part of cloud systems which allows all processors or devices to carry out the same quantity of work in equal time [56]. Load balancing systems are divided into two as static and dynamic [57–60]. Static load balancing algorithms are based on the completion time of a task and the decision-making process of the load distribution is done according to the compile time [56]. In dynamic load balancing algorithms, the decision making process depends on the different characteristics of nodes such as network bandwidth. Dynamic load balancing algorithms are hard to implement since the node needs continuous control [61]. One of the NoSQL databases used in distributed computing systems is Redis. It is a NoSQL database that runs under the open source BSD license and provides high performance [62]. Redis databases have the ability to store many types of data such as string and hash zset [63]. It is generally used by distributed computing systems because of its high-performance database capability and the ability to store data in the server cache [64]. Redis is an open source data storage system preferred by most internet companies due to its ability to save data on server cache [65]. Redis works on an asynchronous storage system by keeping data in memory [66]. It stores data as key-value and quickly retrieves records by taking the word-pair and the timestamp of the insertion frequency as value [67]. By using the smart search feature of Redis, the corresponding time info and word pair can be accessed quickly [68].

Moving from the explanations so far, this study aims to analyze a total of 90 million hotel inquiries, which are gathered with the active use of server side, and the wireless sensor networks (WSN) components as on the background. In detail, the gathered data are sent to different web servers per day by using 4 pcs 128 GB RAM Redis servers and 60 pcs 16 GB RAM web servers. The number of queries sent to web services is about 690 million and the amount of data processed is about 5.9 trillion. Hotel room rates from different web servers were collected by using Redis load balance structure, which is one of the real-time data analysis methods. With the distributing computing architecture, the data is converted to schema format and saved to Redis servers. When the data extraction process from all web services has been completed or when the specified maximum time has been reached, hotel room information saved to the Redis clusters were collected. At this point, a deep autoencoder network was used to optimize the related data. With huge size of the data, when transferring it from server to server or client to server, we need to compress the data size. For the web server communication data is transferred after gzip/deflate compression. This compression reduced the size of data nearly 90%. In this sense, the rooms with the same type or hostel type were grouped and the lowest priced room was kept in the system while the others were removed. Instead of a centralized system to perform these operations, a distributed computing software architecture supported with deep learning was designed according to the density of incoming data and the number of server groups was increased. While the old architecture could not answer 250 thousand queries, the new architecture could answer in 4011 ms. The approach considered in this study is a typical application within the tourism sector. But it can be also applied in different sectors and areas (such as medical-hospitals, e-trade systems, public governmental units) where big data needs to be optimized and analyzed accordingly.

2. Material and method

The following subsections are devoted to the used data, designed system structure and the employed technologies in the context of the solution approach.

2.1. Data and the general system structure

In this study, query data of 200,000 hotels were used within the research process. In detail, data including check-in/check-out dates, number and nationality of people were gathered from 78 different web servers for a company operating in the field of tourism. In order to analyze the data from the servers, Redis as one of the NoSQL databases, HAProxy as load balancer and again Redis for configuration and static data were used. That system was supported with a deep autoencoder network for improving the optimization. Application software was prepared using Asp.Net and C# language with web API.

2.2. Wireless sensor networks and beacons

The data considered in this study was gathered before real-time by communicating considering the related servers and also using wireless sensor networks (WSN) that were designed for real-time tracking of hotel visitors. WSN can be defined as a group of special sensors, which are dispersed spatially for gathering data from the environment [69–71]. Because such sensors are cheap and easy to implement in terms of scalable network structures, there is a great interest in using WSN for different research problems [72–74]. Thanks to WSN models, it is possible to create wide networks where data are passed through nodes and the data is processes or directed by a central point (sink) for specific purposes [75].

In this study, the hotel visitors triggered the beacons, which are a type of wireless sensors. Beacons are known as are small, cheap components[76] that transmit a signal over bluetooth communication. That signal then may be detected by other devices or used by central units for further processes. As beacons have simple electronic architecture and use low energy, they are widely used for tracking actions, analyzing the environment, ensuring navigation and performing interaction based applications in different fields [77–81]. They are also very effective to be used in the tourism sector since they are low-energy devices that can work a few years on one small battery. Furthermore, some lightweight protocols like MQTT has been seen effective to be used along with beacons, in the context of the distributed wireless sensor networks (WSN). Because of that, some small networks of beacons (WSN) were employed within this study in order to deal with real-time data from hotels to the server side. While visitors come to the hotels they reserved, beacons sensed them to make their connections to the WiFi hot spots. When the visitors are confirmed in the context of their e-mail accounts (if their accounts exist on the tourism company side) and their real-time availability in the area of the related hotel, they were asked by the e-mails from the tourism company to give some comments/information about the hotel they're visiting (Here, the beacons with only sensing proximity was used. The customers did not know about existence of the beacons at the hotels and while gathering the data none of personal data was used). WiFi hotspot is the WSN beacon here. When customer connects to hotel's WiFi he enters the email and allow the sending data to the travel company. There is no hidden WSN beacon at this point.

2.3. System for optimization

Nowadays, with the increase of traffic in internet usage, instead of running the software from a centralized system, distributed computing software architectures consisting of distributed nodules are used. The purpose of using distributed computing software architectures is to increase the number of working nodules along with increasing density in internet traffic. The number of nodules that are used can be increased starting with two. In this study, the inquiry process was conducted for the hotel information gathered from the web server of the tourism company. On average, 90 million queries are sent to the server by users daily. Each of the submitted 90 million queries is sent to approximately 42 different web services. This corresponds to an average of 690 million web service queries. Each user query contains an average of 98 hotels, with an average of 16 rooms and hostels data for each hotel. Thus, 90 million (singleton query) $\times$ 42 (average number of providers) $\times$ 98 (average hotel) $\times$ 16 = 5.9 quadrillion hotel price data were collected in total. The room price data of the hotels were analyzed in real time. Figure 1 shows block diagram of the distributed computing system that centralizes the hotel results and then transmits the results to the user by filtering the results. Since the servers communicate with a lot of web services, in case of direct communication between the distributor web server and its suppliers, close to 100 connections are opened on a server. To prevent this, the distrubor web server distributed 100 queries (10 $\times$ 10) to 10 providers web servers, distributing the load in a balanced way.

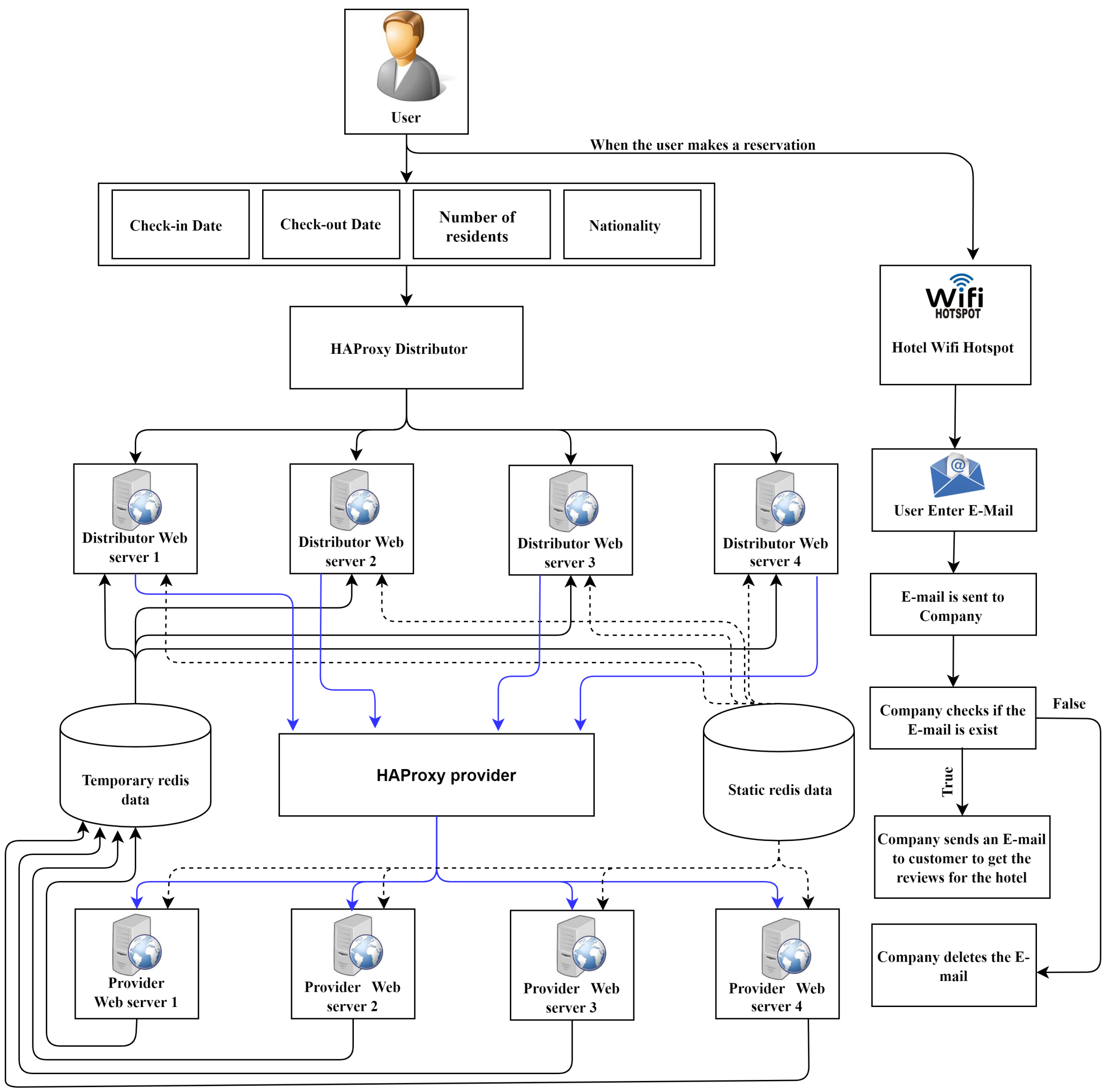

Figure 1. Block diagram of the distributed computing system.

In the developed architecture, requests from the user are sent to the main nodule according to the load status of HAProxy load balancer. Requests received by the service which interprets the data from the user, are distributed to web services through API developed with C#. Web services distribute data to subnodules according to the number of services to analyze. The lower nodule analyzes the data obtained from the main nodule and sends it to the web services. However, there is a need for static data usage when interpreting the data. Instead of retrieving these data from a central database, only the static Redis database used by the lower nodule is used. Thus, using a static Redis database, a new approach has been developed that converts data into a distributed computing logic service instead of using a central database. At this point, a deep autoencoder network was employed for optimizing the data with capabilities of that network type of deep learning. The results obtained from the analysis performed in the lower nodule are interpreted and written to a different Redis database. The software developed for the filtering service transfers the results to the user by singularizing the data read from all subnodules.

As it has an important role on optimization process, some brief information regarding the deep autoencoder network should be given. Generally, autoencoder is an artificial neural network (ANN) model, which can be used for learning about representation of an amount of data, in an unsupervised way [82–84]. Here, numbers of neurons at input and output layers of an autoencoder network are same (Figure 2 [85]). Typically, an autoencoder network learns data encodings in order to ensure dimensional reduction in the processed data. Because of that, optimization of the big data in this study was done with a deep autoencoder network, having a supportive role in the context of the designed system. Autoencoder is one of the most effective techniques of deep learning, which is a subfield of the machine learning. Thanks to deep learning, it is possible to process big amount of data in detail so that more accurate results can be achieved [86–88]. In the era of high automation with intelligent systems, it has been a vital approach to run deep learning against big data. As the big data still has some similarities within, deep learning can be effective to derive optimized patterns [85, 89, 90]. In this study, the gathered big data in the context of tourism sector is directed to the autoencoder taking place within the developed system archictecture. The autoencoder here processed the whole data to reduce the size so that the amount of data, which has the accurate representation is achieved directly.

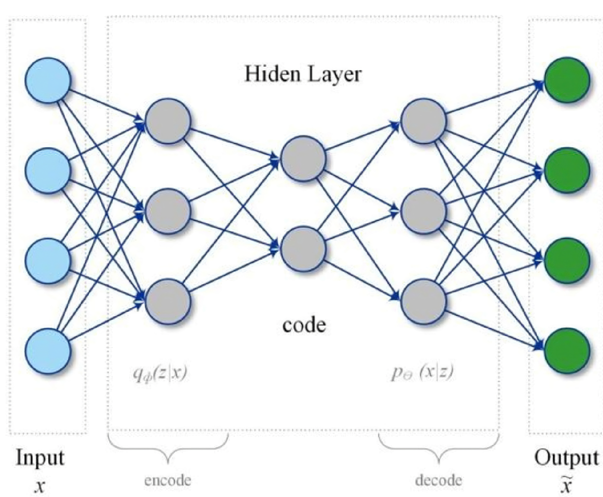

Figure 2. Structure of a deep autoencoder network [82].

3. Application and the findings

In this study, with the software architecture development, the queries were imported from the users via load balancer. A comparison of the response times to the queries of HAProxy and NetScaler as a load balancer is shown in Figure 3.

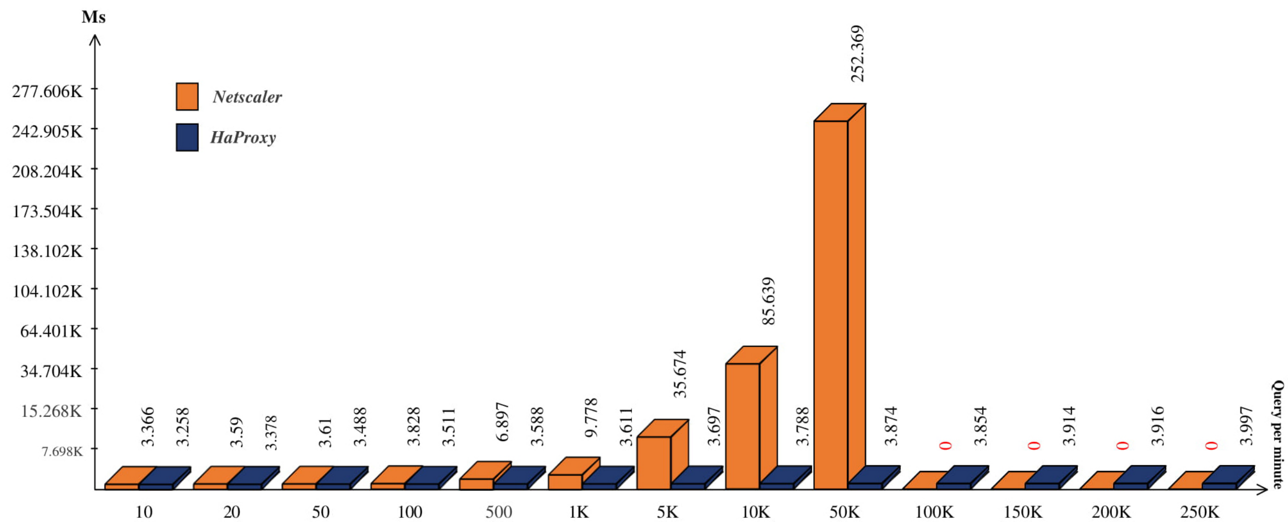

Figure 3. Time comparison of HAProxy and NetScale.

When HAProxy and NetScaler were compared, both HAProxy and NetScaler load balancers performed similar with data sizes up to 100 KB. However, when the size of the data is approximately 1 MB, the response time of the web servers under NetScaler are prolonged. In this case, the web server will not run in an efficient way because there will be clusters in the web servers. It was found that in the same physical environment NetScaler load balancer systems respond later than HAProxy systems in the inquiry transactions with more than 100 queries, and it cannot respond in the inquiry transactions with 50 thousand or more queries. It was also found that systems with HAProxy load balancer respond very quickly to the queries and response times were not affected by the increase in the number of queries. Table 1 shows the response times of the NetScaler and HAProxy load balancer systems to the queries from the web services based on the number of queries per minute.

When the NetScaler and HAProxy response times are examined according to the amount of inquiry per minute, it is seen that both NetScaler and HAProxy perform close to each other until query numbers reaches 100. When the number of queries reached 500–1000 queries per minute, it is seen that the response time of NetScaler increased by 2–3 times compared to HAProxy, and when the number of queries reached from 5 thousand to 50 thousand, the duration time increased by 10 to 80 times, and after 100 thousand query, NetScaler could not answer anymore. It is observed that, HAProxy is barely affected by the increase in the amount of queries and that the duration time is changed little between 3258 ms to 3997 ms. HAProxy was used as a load balancer in the developed software architecture since the amount of the query has little effect on the HAProxy load balancer systems' response times. After performing load balancer operation with HAProxy, requests from users were transferred to the web servers to convert them into the XML or JSON format which are requested by the providers. In the classic system, data is drawn from a central SQL server. With the developed software architecture, the static data was distributed to different Redis clusters that contains four web servers, then the architecture of the classical system and the architecture of the developed software were compared and the results are shown in Figure 4.

Table 1. NetScaler and HAProxy response times per minute.

Query amount (per minute)	NetScaler (ms)	HAProxy (ms)
10	3366	3258
20	3590	3378
50	3610	3488
100	3828	3511
500	6897	3588
1000	9778	3611
5000	35,674	3697
10,000	85,639	3788
50,000	252,369	3874
100,000	N/A	3854
150,000	N/A	3914
200,000	N/A	3916
250,000	N/A	3997

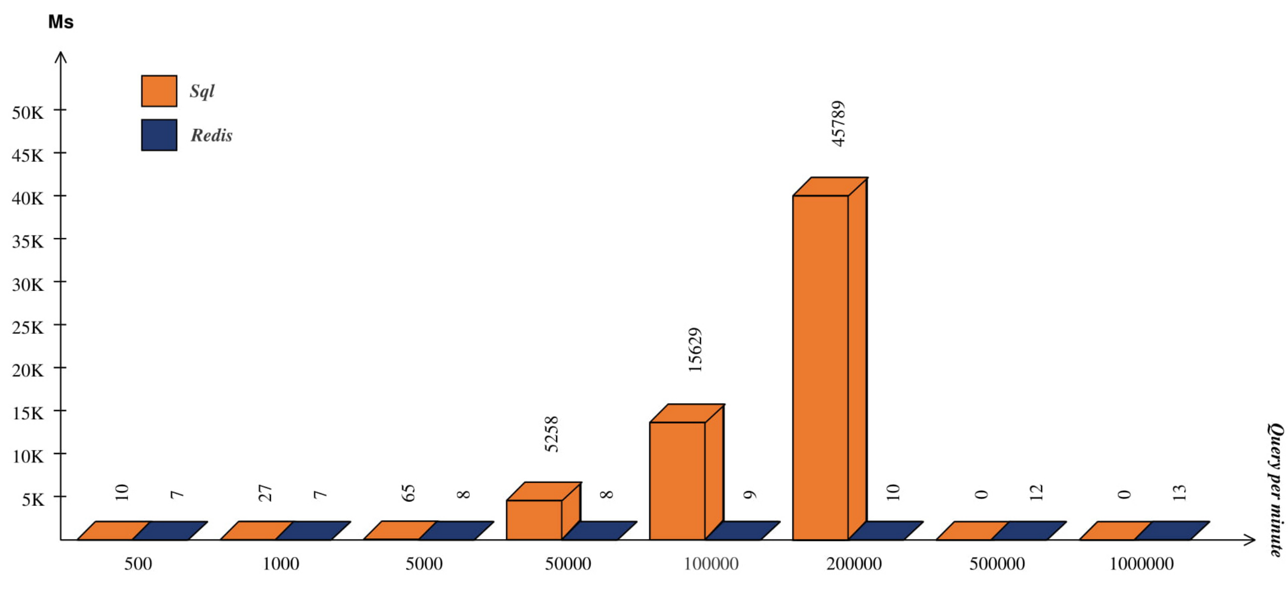

Figure 4. Response times of SQL and Redis servers with an average of 1 MB data per minute.

When the average response time per minute of the SQL and Redis servers to 1 MB data was analyzed, it was seen that the systems that use Redis database are very fast with and over 1000 queries compared to the systems using SQL. Table 2 shows the response times of the SQL and Redis databases with queries range from 50 million to 1 million.

When the response times of SQL and Redis databases are analyzed, it is seen that when the Redis database is used in the developed architecture, the response time varies between 7 to 13 ms when the amount

Table 2. Response times for SQL and Redis databases based on the number of queries per minute.

Query amount (per minute)	SQL (ms)	Redis (ms)
500	10	7
1000	27	7
5000	65	8
50,000	5258	8
100,000	15,629	9
200,000	45,789	10
500,000	N/A	12
1,000,000	N/A	13

of the query increases. However, when SQL server was used, it was seen that there was a very rapid increase in response times between 1000 queries and 200 thousand queries, while there is no response from the server with 500 thousand or more queries.

After the data conversion process is performed on the distributor web servers, the queries are sent to the load balancers of the servers that are going to the provider. Queries received to web services are sent to the providers and the analysis result is obtained by registering to the Redis server of the company. In order to read the data temporarily written to Redis servers, the data of the same type of hotel, room and hostels are grouped and compared according to the amount of data by HashSet and List methods (Figure 5).

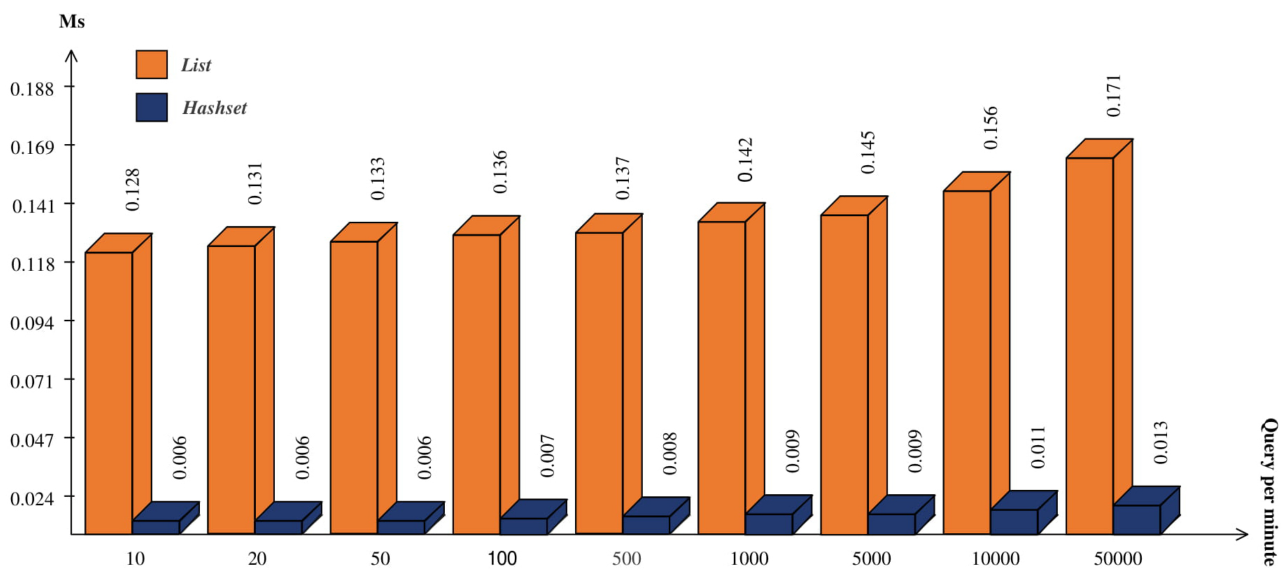

Figure 5. Comparison of List and HashSet according to the amount of data.

When the comparison of List and HashSet according to the amount of data in the Redis servers was examined, it was found that the data reading speed was faster in the HashSet system. The response time of HashSet and List according to the amount of query is given in Table 3.

When HashSet and List data reading systems were examined, it was determined that the response time of HashSet was 13 to 22 times faster than list. Tests were carried out using the same row numbers for both methods using an average of 7500 rows of data. For this reason, HashSet is used for grouping process in the developed software architecture. After performing the HashSet process, the most affordable results are obtained and the other results are deleted from the Redis server.

The response times according to the amount of inquiries per minute of the old architecture that the company has previously used and the newly developed software architecture are shown in Figure 6.

Table 3. HashSet and List response times per minute according to query quantity

Query amount (per minute)	List (ms)	HashSet (ms)
1000	0.128	0.006
2000	0.131	0.006
10,000	0.133	0.006
20,000	0.136	0.007
50,000	0.137	0.008
100,000	0.142	0.009
500,000	0.145	0.009
1,000,000	0.156	0.011
2,000,000	0.171	0.013

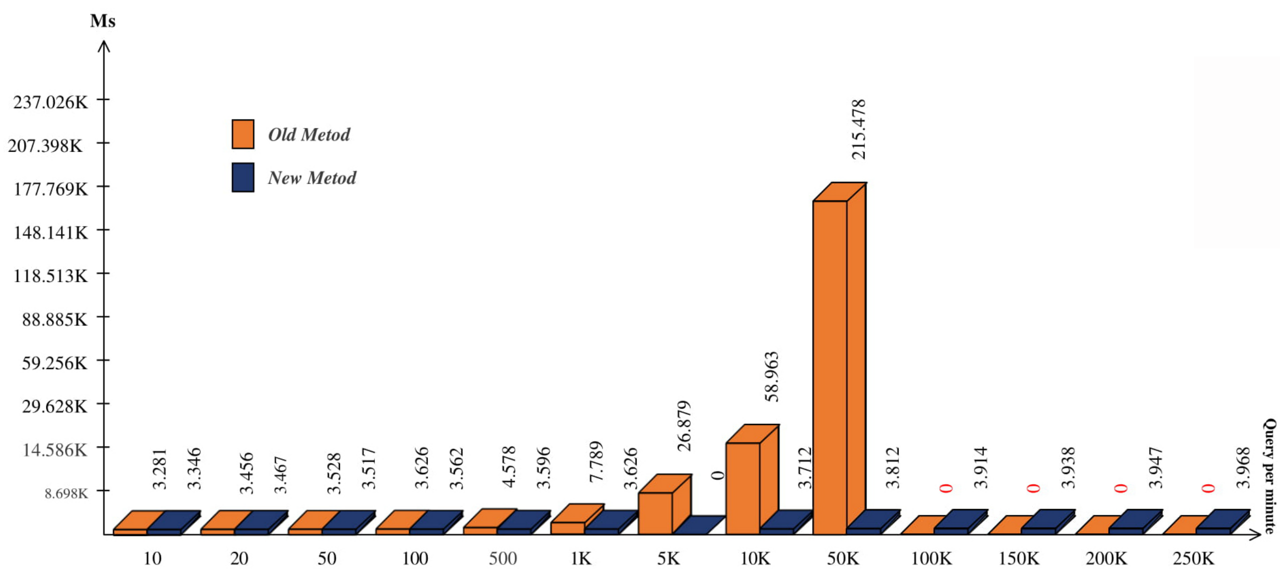

Figure 6. Comparison of the newly developed architecture with the old architecture.

When the old architecture and the developed software architecture were compared, it was found that the response time of both architectures were close to each other up to 500 queries. After 500 queries, it was seen that the old architecture's response time increased and it could not answer over 100 thousand queries. However, with the increase in the amount of queries per minute in the new architecture, it was found that there was a slight change in the response times. In the same hardware environment response time of the newly developed architecture and the old architecture according to the amount of queries per minute is given in Table 4.

As the outcomes, that study has shown that it is a remarkable approach to combine both big data and deep learning for more effective and efficient computational systems of the future. As the humankind

Table 4. Comparison of the newly developed architecture with the old architecture

Query amount (per minute)	Old method (ms)	New method (ms)
10	3281	3346
20	3456	3467
50	3528	3517
100	3626	3562
500	4578	3596
1000	7789	3626
5000	26,879	3712
10,000	58,963	3813
50,000	215,478	3914
100,000	N/A	3938
150,000	N/A	3947
200,000	N/A	3968
250,000	N/A	4011

is experiencing the Industry 4.0 and the era of intelligent systems, findings of this study point the need for optimizing big amount of data and making that by running i.e. encoding-oriented approaches as done in this study. The outcomes of this study also includes showing the potential of the big data-deep learning hybrid systems rising over WSN and distributed software architecture against problems including careful analyze of data with a desired time scope.

4. Conclusion

In this study, 5.9 quadrillion data, gathered from the company operating in the field of reservation services of the tourism sector, were collected from 78 different web servers. Here, wireless sensor networks (WSN) were also used as the triggering element to get the desired data. Instead of the old architecture that was previously used and performed on a single server, a new software architecture that can perform real-time analysis has been developed. In this system, a deep autoencoder network model was used for optimizing the data better. The results obtained with the developed software architecture are given below.

- It has been seen that the distributed software architecture has significantly reduced the response times of the servers.
- It has been found that the developed distributed software architecture in which NetScaler is used as a load balancer could not respond over 100 thousand queries and when it could respond, HAProxy responded 11 times faster than NetScaler.
- It was found that the SQL database used in the old architecture could not respond over 500 thousand queries and when it responds the Redis database that used in the new architecture responds 1160 times faster than the SQL database used in the old architecture.
- In the new architecture, it is found that the HashSet system responds 18 times faster than the List system for reading data in the Redis database.

- When the old architecture and the new architecture were compared, it is found that the old architecture could not respond over 100 thousand queries and when it could respond the new architecture was found to be 9 times faster than the old architecture.

Moreover the related conclusions, the approach considered in this study can be also applied in different sectors where big data needs to be optimized and analyzed effectively. Such sectors and areas may include hospitals, e-trade systems, and public governmental units. Based on that opportunity, further research can include use of the system different sectors-areas. It is also possible to adapt different types of algorithms and/or deep learning techniques to see if capabilities of the solution may be improved. As similar, the authors have already planned future studies for designing alternative autoencoder models for trying to improve findings, and also adapting the same system (whether it is the one in this study or an improve one) in different fields as medical, e-trade systems, and even social media.

Acknowledgments

We would like to thank Perasolutions Yazılım Hizmetleri A.Ş for allowing us to use hotel searching data set.

References

[1] Sokhanvar A. Does foreign direct investment accelerate tourism and economic growth within Europe? Tourism Management Perspectives 2019; 29 (1): 86-96. doi: 10.1016/j.tmp.2018.10.005

[2] Ye BH, Fu H, Law R. Use of impact-range performance and asymmetry analyses to improve OTA website quality. Journal of Hospitality and Tourism Management 2016; 26 (1): 9-17. doi: 10.1016/j.jhtm.2015.09.001

[3] Roudi S, Arasli H, Akadiri SS. New insights into an old issue–examining the influence of tourism on economic growth: evidence from selected small island developing states. Current Issues in Tourism 2019; 22 (11): 1280-1300. doi: 10.1080/13683500.2018.1431207

[4] Chung N, Koo C. The use of social media in travel information search. Telematics and Informatics 2015; 32 (2): 215-229. doi: 10.1016/j.tele.2014.08.005

[5] Okazaki S, Andreu L, Campo S. Knowledge sharing among tourists via social media: a comparison between Facebook and TripAdvisor. International Journal of Tourism Research 2017; 19 (1): 107-119. doi: 10.1002/jtr.2090

[6] Yang Y. Understanding tourist attraction cooperation: an application of network analysis to the case of Shanghai, China. Journal of Destination Marketing & Management 2018; 8 (2): 396-411. doi: 10.1016/j.jdmm.2017.08.003

[7] Xiang Z, Gretzel U. Role of social media in online travel information search. Tourism Management 2010; 31(2): 179-188. doi: 10.1016/j.tourman.2009.02.016

[8] Chu SC, Kim Y. Determinants of consumer engagement in electronic word-of-mouth (eWOM) in social networking sites. International Journal of Advertising 2011; 30 (1): 47-75. doi: 10.2501/IJA-30-1-047-075

[9] Fotiadis AK, Stylos N. The effects of online social networking on retail consumer dynamics in the attractions industry: the case of 'E-da' theme park, Taiwan. Technological Forecasting and Social Change 2017; 124 (11): 283-294. doi: 10.1016/j.techfore.2016.06.033

[10] See-to EW, Ho KK. Value co-creation and purchase intention in social network sites: the role of electronic word-of-mouth and trust–a theoretical analysis. Computers in Human Behavior 2014; 31 (2): 182-189. doi: 10.1016/j.chb.2013.10.013

[11] Filieri R, Alguezaui S, McLeay F. Why do travelers trust TripAdvisor? Antecedents of trust towards consumer-generated media and its influence on recommendation adoption and word of mouth. Tourism Management 2015; 51 (6): 174-185. doi: 10.1016/j.tourman.2015.05.007

[12] Hudson S. Challenges of tourism marketing in the digital, global economy. In: McCabe S (editor). The Routledge Handbook of Tourism Marketing. 1rd ed. London, England: Routledge and CRC Press, 2014, pp. 475-490.

[13] Pantano E, Di Pietro L. From e-tourism to f-tourism: emerging issues from negative tourists' online reviews. Journal of Hospitality and Tourism Technology 2013; 4 (3): 211-227. doi: 10.1108/JHTT-02-2013-0005

[14] Xiang Z, Magnini VP, Fesenmaier DR. Information technology and consumer behavior in travel and tourism: insights from travel planning using the internet. Journal of Retailing and Consumer Services 2015; 22 (1): 244-249. doi: 10.1016/j.jretconser.2014.08.005

[15] Aktan E, Koçyiğit M. Sosyal medyanın turizm faaliyetlerindeki rolü üzerine teorik bir inceleme. Dumlupınar Üniversitesi Sosyal Bilimler Dergisi 2016; 1 (1): 62-73 (in Turkish with an abstract in English).

[16] Zhang Y, Shu S, Ji Z, Wang Y. A study of the commercial application of big data of the international hotel group in China: based on the case study of marriott international. In: IEEE 2015 First International Conference on Big Data Computing Service and Applications; New York, NY, USA; 2015. pp. 412-417. doi:10.1109/BigDataService.2015.59

[17] Qin H, Ye X. Big data analysis based on the hotel online reviews. In: Proceedings of the 2018 3rd International Conference on Automation, Mechanical Control and Computational Engineering; Dalian, China; 2018. pp.1-20. doi: 10.2991/amcce-18.2018.36

[18] Gandomi A, Haider M. Beyond the hype: big data concepts, methods, and analytics. International Journal of Information Management 2015; 35 (2): 137-144. doi: 10.1016/j.ijinfomgt.2014.10.007

[19] Chen CP, Zhang CY. Data-intensive applications, challenges, techniques and technologies: a survey on big data. Information Sciences 2014; 275 (22): 314-347. doi: 10.1016/j.ins.2014.01.015

[20] Rodríguez-Mazahua L, Rodríguez-Enríquez CA, Sánchez-Cervantes JL, Cervantes J, García-Alcaraz JL et al. A general perspective of big data: applications, tools, challenges and trends. The Journal of Supercomputing 2016; 72 (8): 3073-3113. doi: 10.1007/s11227-015-1501-1

[21] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A et al. The rise of "big data" on cloud computing: review and open research issues. Information Systems 2015; 47 (1): 98-115. doi: 10.1016/j.is.2014.07.006

[22] Chen M, Mao S, Liu Y. Big data: a survey. Mobile Networks and Applications 2014; 19 (2): 171-209. doi: 10.1007/s11036-013-0489-0

[23] Wang Y, Kung L, Byrd TA. Big data analytics: understanding its capabilities and potential benefits for healthcare organizations. Technological Forecasting and Social Change 2018; 126 (1): 3-13. doi: 10.1016/j.techfore.2015.12.019

[24] Rathore MMU, Paul A, Ahmad A, Chen BW, Huang B et al. Real-time big data analytical architecture for remote sensing application. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2015; 8 (10): 4610-4621. doi: 10.1109/JSTARS.2015.2424683

[25] Leclerc B, Cale J. Big Data. New York, NY, USA: Routledge and CRC Press, 2020.

[26] Waga D, Rabah K. Environmental conditions' big data management and cloud computing analytics for sustainable agriculture. World Journal of Computer Application and Technology 2014; 2 (3): 73-81. doi: 10.13189/wjcat.2014.020303

[27] Cooper J, Noon M, Jones C, Kahn E, Arbuckle P. Big data in life cycle assessment. Journal of Industrial Ecology 2013; 17 (6): 796-799. doi: 10.1111/jiec.12069

[28] Elhoseny M, Abdelaziz A, Salama AS, Riad AM, Muhammad K et al. A hybrid model of internet of things and cloud computing to manage big data in health services applications. Future Generation Computer Systems 2018; 86 (9): 1383-1394. doi: 10.1016/j.future.2018.03.005

[29] Attaran M, Stark J, Stotler D. Opportunities and challenges for big data analytics in US higher education: a conceptual model for implementation. Industry and Higher Education 2018; 32 (3): 169-182. doi: 10.1177/0950422218770937

[30] Evans MR, Oliver D, Yang K, Zhou X, Ali RY et al. Enabling spatial big data via CyberGIS: challenges and opportunities. In: Wang S, Goodchild M (editors). CyberGIS for Geospatial Discovery and Innovation. Dordrecht, Netherlands: Springer, 2019, pp. 143-170.

[31] Mehta N, Pandit A, Kulkarni M. Elements of healthcare big data analytics. In: Kulkarni AJ, Siarry P, Singh PK, Abraham A, Zhang M et al. (editors). Big Data Analytics in Healthcare. Studies in Big Data, Vol. 66. Cham, Switzerland: Springer Nature Switzerland AG, 2020, pp. 23-43.

[32] Bibri SE. The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability. Sustainable Cities and Society 2018; 38 (3): 230-253. doi: 10.1016/j.scs.2017.12.034

[33] Dezfouli MB, Shahraki MHN, Zamani H. A novel tour planning model using big data. In: 2018 International Conference on Artificial Intelligence and Data Processing; Malatya, Turkey; 2018. pp. 1-6.

[34] Miah SJ, Vu HQ, Gammack J, McGrath M. A big data analytics method for tourist behaviour analysis. Information & Management 2017; 54 (6): 771-785. doi: 10.1016/j.im.2016.11.011

[35] Salas-Olmedo MH, Moya-Gómez B, García-Palomares JC, Gutiérrez J. Tourists' digital footprint in cities: comparing big data sources. Tourism Management 2018; 66 (3): 13-25. doi: 10.1016/j.tourman.2017.11.001

[36] Del Vecchio P, Secundo G, Passiante G. Analyzing big data through the lens of customer knowledge management: evidence from a set of regional tourism experiences. Kybernetes 2018; 47 (7): 1348-1362. doi: 10.1108/K-07-2017-0273

[37] Fuchs M, Höpken W, Lexhagen M. Big data analytics for knowledge generation in tourism destinations–a case from Sweden. Journal of Destination Marketing & Management 2014; 3 (4): 198-209. doi: 10.1016/j.jdmm.2014.08.002

[38] Sun Y, Song H, Jara AJ, Bie R. Internet of things and big data analytics for smart and connected communities. IEEE Access 2016; 4 (1): 766-773. doi:10.1109/ACCESS.2016.2529723

[39] Nguyen TT, Camacho D, Jung JE. Identifying and ranking cultural heritage resources on geotagged social media for smart cultural tourism services. Personal and Ubiquitous Computing 2017; 21 (2): 267-279. doi: 10.1007/s00779-016-0992-y

[40] Boulaalam O, Aghoutane B, El Ouadghiri D, Moumen A, Malinine MLC. Proposal of a big data system based on the recommendation and profiling techniques for an intelligent management of moroccan tourism. Procedia Computer Science 2018; 134 (11): 346-351. doi: 10.1016/j.procs.2018.07.200

[41] Casado R, Younas M. Emerging trends and technologies in big data processing. Concurrency and Computation: Practice and Experience 2015; 27 (8): 2078-2091. doi: 10.1002/cpe.3398

[42] Terzi DS, Demirezen U, Sagiroglu S. Evaluations of big data processing. Services Transactions on Big Data 2016; 3 (1): 44-53. doi: 10.29268/stbd.2016.3.1.4

[43] Huang X, Zhang L, Ding Y. The Baidu index: uses in predicting tourism flows–a case study of the forbidden city. Tourism Management 2017; 58 (1): 301-306. doi: 10.1016/j.tourman.2016.03.015

[44] Zhang YF, Tian YC, Kelly W, Fidge C. Scalable and efficient data distribution for distributed computing of all-to-all comparison problems. Future Generation Computer Systems 2017; 67 (2): 152-162. doi: 10.1016/j.future.2016.08.020

[45] Chou TCK, Abraham JA. Load balancing in distributed systems. IEEE Transactions on Software Engineering 1982; SE-8 (4): 401-412. doi: 10.1109/TSE.1982.235574

[46] Jiang Y. A survey of task allocation and load balancing in distributed systems. IEEE Transactions on Parallel and Distributed Systems 2016; 27 (2): 585-599. doi: 10.1109/TPDS.2015.2407900

[47] Attiya G, Hamam Y. Task allocation for maximizing reliability of distributed systems: a simulated annealing approach. Journal of parallel and Distributed Computing 2006; 66 (10): 1259-1266. doi: 10.1016/j.jpdc.2006.06.006

[48] Jiang Y, Jiang J. Contextual resource negotiation-based task allocation and load balancing in complex software systems. IEEE Transactions on Parallel and Distributed Systems 2008; 20 (5): 641-653. doi: 10.1109/TPDS.2008.133

[49] Liu J, Jin X, Wang Y. Agent-based load balancing on homogeneous minigrids: Macroscopic modeling and characterization. IEEE Transactions on Parallel and Distributed Systems 2005; 16 (7): 586-598. doi: 10.1109/TPDS.2005.76

[50] Rudolph L, Slivkin-Allalouf M, Upfal E. A simple load balancing scheme for task allocation in parallel machines. In: Proceedings of the third annual ACM Symposium on Parallel algorithms and architectures; New York, NY, USA; 1991. pp. 237-245. doi: 10.1145/113379.113401

[51] Chen X, Zhou L. Design and implementation of an intelligent system for tourist routes recommendation based on hadoop. In: 2015 6th IEEE International Conference on Software Engineering and Service Science; Beijing, China; 2015. pp. 774-778. doi: 10.1109/ICSESS.2015.7339171

[52] Jiang SJ, Wu CT, Tseng KH. Utilization of cloud computing app for homestay operation-design and analysis. The International Journal of Organizational Innovation 2018; 11 (2): 35-47.

[53] Sharma S, Singh S, Sharma M. Performance analysis of load balancing algorithms. World Academy of Science, Engineering and Technology 2008; 38 (3): 269-272.

[54] Andrews GR, Dobkin DP, Downey PJ. Distributed allocation with pools of servers. In: Proceedings of the first ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing; Ottawa, ON, Canada; 1982. pp. 73-83. doi: 10.1145/800220.806684

[55] Kumar A, Kalra M. Load balancing in cloud data center using modified active monitoring load balancer. In: International Conference on Advances in Computing, Communication, & Automation; Dehradun, India; 2016. pp. 1-5. doi: 10.1109/ICACCA.2016.7578903

[56] Aslam S, Shah MA. Load balancing algorithms in cloud computing: a survey of modern techniques. In: National Software Engineering Conference; Rawalpindi, Pakistan; 2015. pp. 30-35. doi: 10.1109/NSEC.2015.7396341

[57] Cybenko G. Dynamic load balancing for distributed memory multiprocessors. Journal of Parallel and Distributed Computing 1989; 7 (2): 279-301. doi:10.1016/0743-7315(89)90021-X

[58] Cardellini V, Colajanni M, Yu PS. Dynamic load balancing on web-server systems. IEEE Internet Computing 1999; 3 (3): 28-39. doi: 10.1109/4236.769420

[59] Kim C, Kameda H. An algorithm for optimal static load balancing in distributed computer systems. Transactions on Computers 1992; 41 (3): 381-384. doi:10.1109/12.12745

[60] Tantawi AN, Towsley D. Optimal static load balancing in distributed computer systems. Journal of the ACM 1985; 32 (2): 445-465. doi:10.1145/3149.3156

[61] Zhang J, Ji Y, Song M, Zhao Y, Yu X et al. Dynamic traffic grooming in sliceable bandwidth-variable transponder-enabled elastic optical networks. Journal of Lightwave Technology 2015; 33 (1): 183-191. doi: 10.1109/JLT.2014.2383444

[62] Cattell R. Scalable SQL and NoSQL data stores. Acm Sigmod Record 2011; 39 (4): 12-27. doi: 10.1145/1978915.1978919

[63] Hao R, Guangmin F, Qiang W, Lingxu G, Jian C et al. Research on unified storage technology of multidimensional historical data of substation based on the combination of redis and SSDB. In: 2018 China International Conference on Electricity Distribution; Tianjin, China: 2018. pp. 1251-1255. doi: 10.1109/CICED.2018.8592231

[64] Li S, Jiang H, Shi M. Redis-based web server cluster session maintaining technology. In: 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery; Guilin, China; 2017; pp. 3065-3069. doi: 10.1109/FSKD.2017.8393274

[65] Carlson JL. Redis in Action. Shelter Island, NY, USA: Manning Publications Co., 2013.

[66] Qin J, Ma L, Liu Q. DFTHR: a distributed framework for trajectory similarity query based on HBase and Redis. Information 2019; 10 (2): 77. doi: 10.3390/info10020077

[67] Leea T, Kim Y, Hwang E. Abnormal payment transaction detection scheme based on scalable architecture and redis cluster. In: International Conference on Platform Technology and Service; Jeju, South Korea; 2018. pp. 1-6. doi: 10.1109/PlatCon.2018.8472732

[68] Franciscus N, Ren X, Stantic B. Precomputing architecture for flexible and efficient big data analytics. Vietnam Journal of Computer Science 2018; 5 (2): 133-142. doi: 10.1007/s40595-018-0109-9

[69] Callaway Jr, Edgar H. Wireless Sensor Networks: Architectures and Protocols. Boca Raton, FL, USA: CRC Press, 2003.

[70] Khan S, Pathan ASK, Alrajeh NA. Wireless Sensor Networks: Current Status and Future Trends. Boca Raton, FL, USA: CRC Press, 2016.

[71] Raghavendra CS, Sivalingam KM, Znati T. Wireless Sensor Networks. Rijeka, Croatia: InTech, 2006.

[72] Rehmani MH, Pathan ASK. Emerging Communication Technologies Based on Wireless Sensor Networks: Current Research and Future Applications. Boca Raton, FL, USA: CRC Press, 2016.

[73] Randhawa S, Jain, S. Data aggregation in wireless sensor networks: previous research, current status and future directions. Wireless Personal Communications 2017; 97 (3): 3355-3425. doi: 10.1007/s11277-017-4674-5

[74] Elayan H, Shubair RM, Kiourti A. Wireless sensors for medical applications: current status and future challenges. In: 11th European Conference on Antennas and Propagation; Paris, France: 2017. pp. 2478-2482. doi: 10.23919/EuCAP.2017.7928405

[75] Zhang X, Dong L, Peng H, Chen H, Zhao S et al. Collusion-aware privacy-preserving range query in tiered wireless sensor networks. Sensors 2014; 14 (12): 23905-23932. doi: 10.3390/s141223905

[76] Kemmerling M, Schliepkorte H, Duisburg F. An orientation and information system for blind people based on RF-speech-beacons. In: TIDE 3th Congress; Helsinki, Finland; 1998. pp.1-20.

[77] Kose U, Vasant P. Better campus life for visually impaired University students: intelligent social walking system with beacon and assistive technologies. Wireless Networks 2018; 24 (8): 1-15. doi: 10.1007/s11276-018-1868-z

[78] Antevski K, Redondi AE, Pitic RA. Hybrid BLE and Wi-Fi localization system for the creation of study groups in smart libraries. In: 9th IFIP Wireless and Mobile Networking Conference (WMNC); Colmar, France; 2016. pp. 41-48. doi: 10.1109/WMNC.2016.7543928

[79] Suda Y, Arai T, Yoshizawa T, Fujita Y, Zempo K et al. Shopping baskets for on-line beacon sensor network in retail store. In: 16th IEEE Annual Consumer Communications & Networking Conference (CCNC); Las Vegas, NV, USA; 2019. pp. 1-2. doi: 10.1109/CCNC.2019.8651857

[80] Al-Kaisy A, Miyake GT, Staszcuk J, Scharf D. Motorists' voluntary yielding of right of way at uncontrolled midblock crosswalks with rectangular rapid flashing beacons. Journal of Transportation Safety & Security 2018; 10 (4): 303–317. doi: 10.1080/19439962.2016.1267827

[81] Molapo NA, Malekian R, Nair L. Real-time livestock tracking system with integration of sensors and beacon navigation. Wireless Personal Communications 2019; 104 (2): 853-879. doi: 10.1007/s11277-018-6055-0

[82] Sun W, Shao S, Zhao R, Yan R, Zhang X et al. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. Measurement 2016; 89 (13): 171-178. doi: 10.1016/j.measurement.2016.04.007

[83] Su Y, Li J, Plaza A, Marinoni A, Gamba P et al. DAEN: Deep autoencoder networks for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing 2019; 57 (7): 4309-4321. doi: 10.1109/TGRS.2018.2890633

[84] Xiong Y, Zuo R. Recognition of geochemical anomalies using a deep autoencoder network. Computers & Geosciences 2016; 86 (1): 75-82. doi:10.1016/j.cageo.2015.10.006

[85] Mohammadi M, Al-Fuqaha A, Sorour S, Guizani M. Deep learning for IoT big data and streaming analytics: a survey. IEEE Communications Surveys & Tutorials 2018; 20 (4): 2923-2960. doi: 10.1109/COMST.2018.2844341

[86] Hosseini MP, Lu S, Kamaraj K, Slowikowski A, Venkatesh HC. Deep Learning Architectures. In: Deep Learning: Concepts and Architectures. Cham, Switzerland: Springer, 2020, pp. 1-24.

[87] Goodfellow I, Bengio Y, Courville A. Deep Learning. London, England: MIT Press, 2016.

[88] Buduma N, Locascio N. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms. Sebastopol, CA, USA: O'Reilly Media Inc., 2017.

[89] Zhang Q, Yang LT, Chen Z, Li P. A survey on deep learning for big data. Information Fusion 2018; 42 (1): 146-157. doi: 10.1016/j.inffus.2017.10.006

[90] Sohangir S, Wang D, Pomeranets A, Khoshgoftaar TM. Big data: deep learning for financial sentiment analysis. Journal of Big Data 2018; 5 (1): 1-25. doi: 10.1186/s40537-017-0111-6