I.J. Information Technology and Computer Science, **2021, 4, 47-52**
Published Online August 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2021.04.05

SBIoT: Scalable Broker Design for Real Time Streaming Big Data in the Internet of Things Environment

Halil ARSLAN
Sivas Cumhuriyet University/Computer Engineering Department, Sivas, 58140, Turkey
E-mail: harslan@cumhuriyet.edu.tr

Mustafa YALCIN and Yasin ŞAHAN
Detaysoft/ E-Solution Manager, İstanbul, Turkey
E-mail: {mustafa.yalcin, yasin.sahan}@detaysoft.com

Received: 27 January 2021; Revised: 05 March 2021; Accepted: 18 March 2021; Published: 08 August 2021

Abstract: Thanks to the recent development in the technology number of IoT devices increased dramatically. Therefore, industries have been started to use IoT devices for their business processes. Many systems can be done automatically thanks to them. For this purpose, there is a server to process sensors data. Transferring these data to the server without any loss has crucial importance for the accuracy of IoT applications. Therefore, in this thesis a scalable broker for real time streaming data is proposed. Open source technologies, which are NoSql and in-memory databases, queueing, full-text index search, virtualization and container management orchestration algorithms, are used to increase efficiency of the broker. Firstly, it is planned to be used for the biggest airport in Turkey to determine the staff location. Considering the experiment analysis, proposed system is good enough to transfer data produced by devices in that airport. In addition to this, the system can adapt to device increase, which means if number of devices increasing in time, number of nodes can be increased to capture more data.

Index Terms: Internet of Things, Big Data, Broker, Scaling, IoT.

1. Introduction

Recently, the concept of the Internet of Things (IoT) increasing its popularity with the rapid development of technology. IoT applications are used in almost every field such as finance, media, informatics and education. Therefore, number of IoT devices the number of IoT devices reached enormous values. Consequently, a large amount of data has been produced. Making useful applications using different kind of sensors such as bluetooth, temperature, humidity and location has important roles for the business processes of companies. Because of these reasons researcher applied many useful applications using IoT devices. [1–4]. They want to make autonomous systems with sensors instead of using human power. In such applications, sensors transmit the data periodically to a server to transform them into meaningful applications. Because there are too many devices, traditional vertical applications are not sufficient for a lossless transmission. As well as data loss, server response time also increasing dramatically [5]. Therefore, a lot of research and development has been done in the industry on NoSql based Broker.

Although there have been many studies in this area, collecting and analyzing real-time big data still appears as an unresolved problem. There are some existing solutions in the industry but to catch data from any protocols may be problem for them. In addition to this, they are not using micro-services, therefore it will be hard to adapt to the new systems [6]. In this study, a scalable broker for data that overserved from IoT devices was proposed to solve explained problems. This broker supports four types of transfer protocols: TCP, UDP, HTTP and MQTT. Because this system developed using micro-service architecture, a new transfer protocol can be integrated the system easily. Virtualization and container orchestration techniques are used to make system scalable. In addition to these, the broker contains optional signal processing module to make signals more robust.

2. Related Works

Broker industry capital, which has an important place in the industry, is at the level of billions of dollars worldwide [7]. In addition to industry, many studies have been done in the academic field. Manasrah et al. proposed variable service broker routing policy to select optimal data center and they showed that proposed system has the best response

and data processing time [8]. Cheng et al. design a broker on CIDAP platform for the data, which collected from smart cities [9]. Dobre and Xhafa proposed a data collection and storing center for big data which, uses contextual information system architecture [10]. Nguyen et al. extract the advantages and disadvantages of Apache ActiveMQ and Kafka, which are open source technologies, by comparing the characteristic features of them [11]. Jutadhamakorn et al. develop scalable Message Queuing Telemetry Transport (MQTT) broker by using low-cost computers and open source broker technologies [12]. Kawaguchi and Bandai proposed a broker for location based data and they used theoretical evaluation to show effectiveness of it [13]. Raj and Kumar introduced a node sequencing based model to convert XML twig query into a branch sequence, which can transfer tens of thousands data [14]. Rathod et al. implement publish-subscribe based model to handle data transfer between the SCADA components [15]. Kelemen analysis integration message broker models for the airport information system industry [16]. Badidi described a message broker platform to deal with heterogeneity of IoT devices and to scale system according to data volume [17]. Pipatsakulroj et al. developed high-performance MQTT broker that run on single machine by efficiently utilizing multi-core CPUs [18].

3. Methods

3.1. RabbitMQ

The biggest problem of systems with excessive data flow and transactions is the order in which data will be sent and processed. Especially in the systems that has too much and frequent real time data, one of the biggest issues, which effects the server response time, is processing the data asynchronously. Too many data from different kind of IoT sensors will be transferred the proposed system. Data came from these sensors need to be stored in database and processed in parallel. RabbitMQ [19], which is open source message queueing algorithm, is used to solve mentioned problem. RabbitMQ advantages can be listed as follows:

- It supports the different kind of protocols
- It provides high performance
- Easy integration
- Message delivery guarantee

3.2. Data Storage Tools

Considering the number of devices and transmission frequency, data size can increase dramatically. For this reason, using relational databases causes the increasing in the server response time. Therefore, NoSql databases, which store data in json format, commonly used in big data applications. Another important feature of NoSql databases, which work much faster than relational databases, is that they can be scalable horizontally. In this study, MongoDB [20], which is open source NoSQL database, is used.

Although NoSQL databases is much faster than relational databases, they are sometimes not enough for real time big data processing. In memory databases using as an alternative to solve this problem. These databases keep data in the memory instead of storing them into solid-state disk. Therefore, data access delay can be computed in microseconds. In this study Redis [21], which supports many multipurpose data structures, is used as an in-memory database.

Some applications make search for same element many times such as last n data for same sensors in each second. Because database contains many data from many sensors, to make search in all databases will be ineffective for such cases. Generating small parts by indexing the target element decreases time to access data. Full-text index search technique is used to solve this problem. In this study, ElasticSearch [22], which is open source full text index tool, is used

3.3. Docker

Running more than one operating system on the same physical device is called virtualization. The proposed system needs virtualization in order to work platform-independent and to integrate with the third part systems easily. In this study, Docker [23] technology has been used as a virtualization technology. Docker is a fully open source virtualization platform that has a more flexible structure than other virtual machines. Unlike virtual machines, it provides isolation of applications using container structures on a single operating system. A system created with Docker can only use the resources allocated to itself and can share resources at a minimal level. Image and container concepts start to using frequently with Docker. Image is called the structure of the operating system, programs and libraries required to run the target application. On the other hand, container, on the is called the operated or stopped state of the created Images. Thanks to the Docker, a program made with these features can be easily distributed and resource usage can be set. Containers created with Docker can work in isolation from each other. Therefore, two applications, which need different versions of the same program, can be run on the same physical machine. For example, two different applications that require Python 3 and Python 2 can run two separate containers and run on the same physical machine in a way that can be integrated with each other.

3.4. Kubernetes

Applications with high real-time data flow need to be scalable horizontally, so they can overcome to transfer massive data. This method is a good solution for instant data flow analysis, but the load balance between duplicate applications needs to be well established [24]. Load balance can be briefly defined as the process of equally distributing incoming transaction traffic to duplicate applications. In this study, several copies of proposed broker are created using Docker architecture and load balance between the containers provided by Kubernetes [25].

Kubernetes is a container management system first used by Google, then open source to everyone. Nowadays, It attracts attention as one of the most used container management systems. Basically, Kubernetes has a master node and it is not preferred to run jobs on this main node. There are several more nodes connected to the master node and workloads are distributed among these nodes. If desired, the system can be designed for higher use by determining in more than one master node. Kubernetes not only provides the workload between nodes, but also enables the node to become reusable in case of a problem with any of the nodes. Kubernetes advantages can be listed as follows:

- The entire structure can be seen as a single organization
- More efficient resources usage
- Health check for containers
- Transition to the new version without interruption
- Ensuring that containers work in an isolated network

3.5. Signal Processing

Although, IoT devices are more advance than the past, data taken from them may contain many noise because of the environmental effects. Cleaning that noises from data have crucial importance for many application, which are uses IoT devices. For example, bluetooth low energy, which commonly used indoor positioning applications, can effected from environmental things easily. Determining the instant true data for given bluetooth low energy device effect the accuracy of position. For this reason, optional signal processing module is added to proposed broker. This part contains 3 different types of techniques: Taking mean of last n data for given device, Kalman filters [26], and Butterworth filters proposed by Mahmud et al [27]. Detailed information about these filters can be found in the related papers.

4. Proposed System and Load Test

In this study, a broker was designed to help store and process data from IoT devices for situations where real-time and instant data flow is too high. Devices can be forwarded with the four different types of transfer protocols: TCP, UDP, HTTP and MQTT. Firstly, producer layer of Broker catches data taken from IoT devices and they are posted the queuing layer. In this layer, sensor data queued thanks to RabbitMQ. Third layer of the system is called as broker consumer that is listen the queue. Consumer takes data from queue and it simultaneously save raw data to MongoDB and transmit raw data to data processor layer, which is the layer for transforming raw data to meaningful information such as rssi value of sensor, temperature value of sensor, mac address of device etc. After parsing the data, required fields for application can be stored in MongoDB or Redis or ElasticSearch can index them. Selecting one or more of these store types depends on application to application. Therefore, proposed broker will have 4 different web service: post service for raw data in MongoDB, post service for parsed data in MongoDB, post service for ElasticSearch and post service for parsed data in Redis. For three of them, which are contains parsed data, users can choose to use signal processing signal processing is selected, also method should be selected. Then, the result of service will be taken by the signal processing method. All layer in this system is virtualized using Docker technology and more than o copies for each were generated. Kubernetes architecture was used to balance the load between these copies. Due to Kubernetes requirements, proposed system should contain at least three copy of each application. The three copies of this proposed structure are summarized in Figure 1.

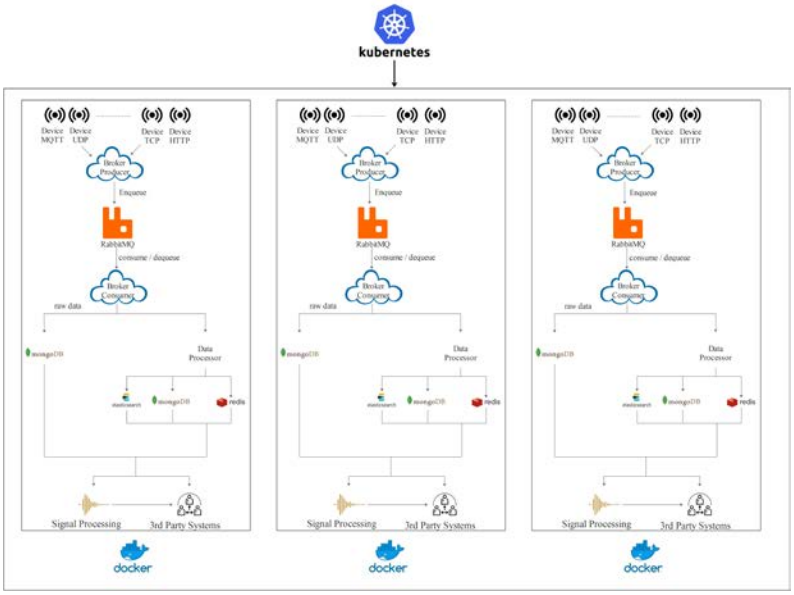

Fig.1. Proposed System Architecture

Since the number of data collected in this system is too high, it increases cumulatively over time. Therefore, jobs to clean expired data for each storage system were developed. Expire time for each application and each storage can be selected separately. It's recommended that expire time for ElasticSearch and Redis should be low. One the one hand, historical data is stored in the MongoDB, so it can be easily accessible when it is necessary. On the other hand, short-term data can be stored in faster storage technologies, so it can be processed faster.

In order to perform a load test on the proposed system, kubernetes systems containing 1 master node and 3, 5, 7 and 9 worker nodes respectively on 2.3 ghz 4 core 4 machines with 16gb ram per each were installed and how much data the installed systems can process per second was tested. Table 1 shows the analysis results obtained for each node in the hand. For this purpose, a client developed asynchronously to send fake sensor data to system for each protocol types including TCP, UDP, HTTP and MQTT. Total data that sent to system is equally divided between these protocols. Each client contains fake beacon data with random RSSI value between the -65 and -90. Initially, 500 sensors data per seconds sent to system, and this number has been increased by 500. Data transmission was stopped when broker start to loss data or start to use 90 percent of processors and number of maximum data per second is noted. This process was done for system that do not use signal processing, system that use signal processing, and system which have both data need signal processing or not. In addition to this, storage technologies and transfer protocols are used equally, while testing

Table 1. Proposed System Load Test (Number of Messages per Seconds)

Number of Workers	System without signal processing	System with signal processing	Combination of both
3	65000	42500	52500
5	101000	61000	81000
7	131500	80500	106500
9	159500	106500	125500

For each system in table 1, where each of them has different number of nodes, load test repeated 10 times. In each time the maximum number of data transferred varies and they are averaged to calculate values in table 1. In addition to this standard deviations, which are smaller than 1% of mean maximum number of data, were calculated. Considering these information, it can be said that these results are reliability and accurate. This system was designed to use in airport, which is the biggest airport in Turkey, for indoor localization system that calculate the position using bluetooth low energy. In this system, the positions of the employees are desired to be determined with beacon. For this purpose, 2000 gateways were placed at 30 meter intervals. These gateways scan beacons and transmit data to the proposed broker for each two seconds. Each employee has a card and it is thought that 35000 employees will have these cards in total. Each card can be seen by an average of 5-6 gateways at the same time. Considering all these, the designed broker must be able to process 105000 data per second. According to the experiment results, our broker can meet this number of messages with 4 machines even if signal processing is using. Comparing to the other systems [6, 28], our platform transferring at least 10% more data per seconds with the same resources. In addition to this, if resources are evaluated according to prices and the amount of total work, it is also suitable for the Airport applications.

5. Conclusions

In this study, a broker designed to store and process the real time streaming IoT devices data. The proposed system will be used in an airport in the Turkey to process sensor data that come from several type of devices. Before broker design, field of the airport tested and the data flow that may occur has been analyzed. After that, load testing was done to check if this broker is proper for this airport. According to the analysis, proposed broker can handle data flow in this airport easily. Additionally, proposed system can easily update and easily integrated with any system thank to the virtualization. Besides all these, it is thought that this broker will provide the infrastructure of machine learning projects as future studies. Thanks to its ability to collect a lot of data, it will enable us to obtain training data in a short time, as well as its speed in instant transactions, allowing the trained models to produce fast results on live data. Because this broker can adapt to device increase, transferring many data in seconds and it supports to different protocols, it can be used in many IoT applications as an intermediate later. Existing brokers do not have option to whether applying signal processing or not. That's why, this study is novel in Internet of Things field.

Acknowledgment

This study is a output of studies conducted in DetaySoft research and development center. We appreciate their support. The numerical calculations reported in this paper were fully/partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

References

[1] Tahmasib Fataliyev, Shakir Mehdiyev. "Industry 4.0: The Oil and Gas Sector Security and Personal Data Protection", International Journal of Engineering and Manufacturing, Vol.10, No.1, pp.1-14, 2020.
[2] Audrius Urbonavicius, Nagham Saeed. " IoT Leak Detection System for Building Hydronic Pipes ", International Journal of Engineering and Manufacturing, Vol.9, No.5, pp.1-21, 2019.
[3] Palle Divya Vani, Kanchi Raghavendra Rao, "Implementation of Smart Agriculture using CloudIoT and its Geotagging on Android Platform", International Journal of Engineering and Manufacturing, Vol.9, No.2, pp.43-53, 2019.
[4] Syed Kashan Ali Shah, Waqas Mahmood, " Smart Home Automation Using IOT and its Low Cost Implementation ", International Journal of Engineering and Manufacturing, Vol.10, No.5, pp.28-36, 2020.
[5] Yiming Xu, V. Mahendran, and S. Radhakrishnan, 'Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery', in 2016 8th International Conference on Communication Systems and Networks (COMSNETS), Jan. 2016, pp. 1–6, doi: 10.1109/COMSNETS.2016.7439974.
[6] 'Thinger.IO', Thinger.IO. https://console.thinger.io/#!/login.
[7] L. Roderick, 'Discipline and Power in the Digital Age: The Case of the US Consumer Data Broker Industry', Crit. Sociol., vol. 40, no. 5, pp. 729–746, Sep. 2014, doi: 10.1177/0896920513501350.
[8] A. M. Manasrah, T. Smadi, and A. ALmomani, 'A Variable Service Broker Routing Policy for data center selection in cloud analyst', J. King Saud Univ. - Comput. Inf. Sci., vol. 29, no. 3, pp. 365–377, Jul. 2017, doi: 10.1016/j.jksuci.2015.12.006.
[9] B. Cheng, S. Longo, F. Cirillo, M. Bauer, and E. Kovacs, 'Building a Big Data Platform for Smart Cities: Experience and Lessons from Santander', in 2015 IEEE International Congress on Big Data, Jun. 2015, pp. 592–599, doi: 10.1109/BigDataCongress.2015.91.
[10] C. Dobre and F. Xhafa, 'Intelligent services for Big Data science', Future Gener. Comput. Syst., vol. 37, pp. 267–281, Jul. 2014, doi: 10.1016/j.future.2013.07.014.
[11] C. N. Nguyen, J. Lee, S. Hwang, and J.-S. Kim, 'On the role of message broker middleware for many-task computing on a big-data platform', Clust. Comput., vol. 22, no. 1, pp. 2527–2540, Jan. 2019, doi: 10.1007/s10586-018-2634-9.
[12] P. Jutadhamakorn, T. Pillavas, V. Visoottiviseth, R. Takano, J. Haga, and D. Kobayashi, 'A scalable and low-cost MQTT broker clustering system', in 2017 2nd International Conference on Information Technology (INCIT), Nov. 2017, pp. 1–5, doi: 10.1109/INCIT.2017.8257870.
[13] R. Kawaguchi and M. Bandai, 'A Distributed MQTT Broker System for Location-based IoT Applications', in 2019 IEEE International Conference on Consumer Electronics (ICCE), Jan. 2019, pp. 1–4, doi: 10.1109/ICCE.2019.8662069.
[14] A. Raj and P. S. Kumar, 'Branch Sequencing Based XML Message Broker Architecture', in 2007 IEEE 23rd International Conference on Data Engineering, Apr. 2007, pp. 656–665, doi: 10.1109/ICDE.2007.367911.
[15] P. Rathod, D. Sharma, and A. Golhani, 'A topic-based publish-subscribe message broker for SCADA system using hierarchical subscription handling', in 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), Dec. 2017, pp. 1–10, doi: 10.1109/ICAC3.2017.8318758.
[16] Z. Kelemen, 'Airport information system integration by using message broker', Period. Polytech. Transp. Eng., vol. 37, no. 1–2, Art. no. 1–2, Nov. 2009, doi: 10.3311/pp.tr.2009-1-2.03.
[17] E. Badidi, 'Towards a Message Broker Based Platform for Real-Time Streaming of Urban IoT Data', in Computational and Statistical Methods in Intelligent Systems, Cham, 2019, pp. 39–49, doi: 10.1007/978-3-030-00211-4_5.
[18] W. Pipatsakulroj, V. Visoottiviseth, and R. Takano, 'muMQ: A lightweight and scalable MQTT broker', in 2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Jun. 2017, pp. 1–6, doi: 10.1109/LANMAN.2017.7972165.

[19] 'RabbitMQ', RabbitMQ, 2020. https://www.rabbitmq.com/.
[20] 'MongoDB', MongoDB, 2020. https://www.mongodb.com/.
[21] 'Redis', Redis, 2020. https://redis.io/.
[22] 'ElasticSearch', ElasticSearch, 2020. https://www.elastic.co/.
[23] 'Docker', Docker. https://www.docker.com/.
[24] D. Bernstein, 'Containers and Cloud: From LXC to Docker to Kubernetes', IEEE Cloud Comput., vol. 1, no. 3, pp. 81–84, Sep. 2014, doi: 10.1109/MCC.2014.51.
[25] 'Kubernetes', Kubernetes, 2020. https://kubernetes.io/.
[26] R. Chen and J. S. Liu, 'Mixture Kalman filters', J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 62, no. 3, pp. 493–508, 2000, doi: https://doi.org/10.1111/1467-9868.00246.
[27] M. A. Mahmud, A. Abdelgawad, K. Yelamarthi, and Y. A. Ismail, 'Signal processing techniques for IoT-based structural health monitoring', in 2017 29th International Conference on Microelectronics (ICM), Dec. 2017, pp. 1–5, doi: 10.1109/ICM.2017.8268825.
[28] 'Thingstream', Thingstream. https://portal.thingstream.io/?returnUrl=%2Fapp%2Fshop.

Authors' Profiles

Halil Arslan received BS, MS and PhD, degree in electronic and computer education from Sakarya University, Turkey in 2004-2015 respectively. Since 2017, he has been teaching software engineering and computer networks courses as an assistant professor in the computer engineering department. His research interests are computer networks, cyber security and software engineering.

Mustafa Yalcın received the bachelor's degree in computer engineering department from the Karadeniz Technical University. Currently, he is working in Detaysoft as a product solution manager. In these days, he is working on projects, which are performance improvement, business sequence process, integration between systems and encryption and system.

Yasin Şahan Mr. ŞAHAN graduated from Cumhuriyet University Management Information Systems Department and started to work as a web developer in Detaysoft Company in 2011. He has been working as a team manager in the E-product department in 2018 and as a department manager in the e-product department since 2020.

How to cite this paper: Halil ARSLAN, Mustafa YALCIN and Yasin ŞAHAN, "SBIoT: Scalable Broker Design for Real Time Streaming Big Data in the Internet of Things Environment", International Journal of Information Technology and Computer Science(IJITCS), Vol.13, No.4, pp.47-52, 2021. DOI: 10.5815/ijitcs.2021.04.05