ICIC Express Letters
Part B: Applications
Volume **14**, Number **12**, December **2023**

ICIC International ©2023 ISSN 2185-2766
pp. 1295–1305

SMART SLEEP MONITORING SYSTEM BASED ON MICROSERVICES ARCHITECTURE AND EDGE COMPUTING

David Jayaatmaja[1], Stanley Dave Teherag[1], Nico Surantha[1,2,*]
and Sani Muhamad Isa[1]

[1]Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ david037; stanley.teherag }@binus.ac.id; sisa@binus.edu
*Corresponding author: nico.surantha@binus.ac.id

[2]Department of Electrical, Electronic and Communication Engineering
Faculty of Engineering
Tokyo City University
Setagaya-ku, Tokyo 158-8557, Japan

Received February 2023; accepted May 2023

Abstract. *Internet of Things (IoT) based sleep monitoring is improving and can be found easily nowadays, but it depends on a good network environment to work properly. In this study, an edge computing-based smart sleep monitoring system was built by migrating a cloud computing-based smart sleep monitoring system into an edge device. The system was built based on the edge computing concept, which is built inside the edge device and based on microservices architecture. The input for the cloud computing-based sleep monitoring system was Electrocardiogram (ECG) data. The experimental result indicates that the average latency and response time on an edge computing-based system decreased by 27.04% and 33.81% respectively, compared to the cloud computing-based system, but the edge computing-based system's average throughput was slower by 15.08% compared to the cloud computing-based system.*

Keywords: Internet of Things, Microservices, Health monitoring, Edge computing, Smart sleep monitoring

1. **Introduction.** There are lots of studies that propose the definition of health in this era, for example, [1-3]. However, among all of them, World Health Organization (WHO) stated a definition that is the foundation and brings huge impact for the development of medical studies in 1948. "A state of complete physical, mental and social well-being and not merely the absence of disease or infirmity" [4] is the definition that WHO stated. Society highly valued and prioritized health as an outcome and a state of being [5]. It is an asset that people need for contributing in society [6]. Especially in an era of diseases, many people start to realize that they need to pay more attention to their health. Sleep is very important for health and well-being in human's life. It can reduce the risk of accidents and injuries caused by sleepiness and fatigue. A healthy sleep needs to fulfill certain requirements, such as sufficient sleep duration, good sleep timing, regularity of sleep, and absence of sleep disorders [7,8]. There are already many sleep monitoring systems developed for this problem, but most of them are cloud-based systems that depend on a good network environment, for example, [9-12]. With the current advancements in sensor technology and information technology, it is possible to monitor someone's sleep data accurately by using the Internet of Things (IoT) that included sensors, cloud and mobile technologies. Many IoT systems used cloud to perform calculations and other processes

---

DOI: 10.24507/icicelb.14.12.1295

that lead to slower response time because of cloud computing's delay on a large-scale transmission and demanding requirements for load and process capacity on cloud [13]. When it comes to someone's life, accuracy is not the only crucial factor. In the medical world, just a split-second difference can lead to very different results. That is why it should not only be accurate, but it also needs to be able to work on all places including places that have poor network environment. To do that, there are some requirements that need to be fulfilled such as the latency, throughput, and response time. Processing the collected data locally can reduce the computational load on the central cloud server and is particularly useful in situations where time delay is critical, such as in healthcare, since it reduces the need to transmit large amounts of data over the network [14]. The aims for this study are

1) IoT architecture for sleep monitoring system that runs on edge device;
2) Sleep monitoring system that runs on edge device;
3) Performance analysis of edge computing-based sleep monitoring system.

In this study, the focus is on improving the throughput and reducing the response time and latency of the healthcare monitoring system, while ensuring the privacy of users' sensitive data such as sleep data [15]. This will be achieved by using a combination of cloud computing and edge computing. The computational process will be carried out on edge devices, making it possible to perform computations even in poor network environments.

Stable performance in latency, response time, and throughput are aimed to be achieved in this work. The benefit from the proposed system for health monitoring providers is that they can provide a service that can work well on poor network environments [16], especially for services that are sensitive to latency and response time [17]. The structure of this paper is organized as follows. Section 2 provides works that have been done previously that have related subject on health monitoring, IoT, edge computing and cloud computing. The proposed architecture and cloud-to-edge migration steps are presented on Section 3. Section 4 presents the performance analysis of the proposed architecture. Section 5 provides conclusion and future works that can be done.

2. **Related Work.** Several previous works related to healthcare monitoring systems are discussed, including a cloud-based sleep monitoring system proposed by Surantha et al. [9], the use of edge computing to capture human states during a pandemic [19], a system that combines an Arduino UNO-based system with cloud computing for critical care of COVID-19 patients [20], a cloud-based model for monitoring student health [21], and the use of deep learning and cloud computing to predict heart disease from sensor data and medical records [22].

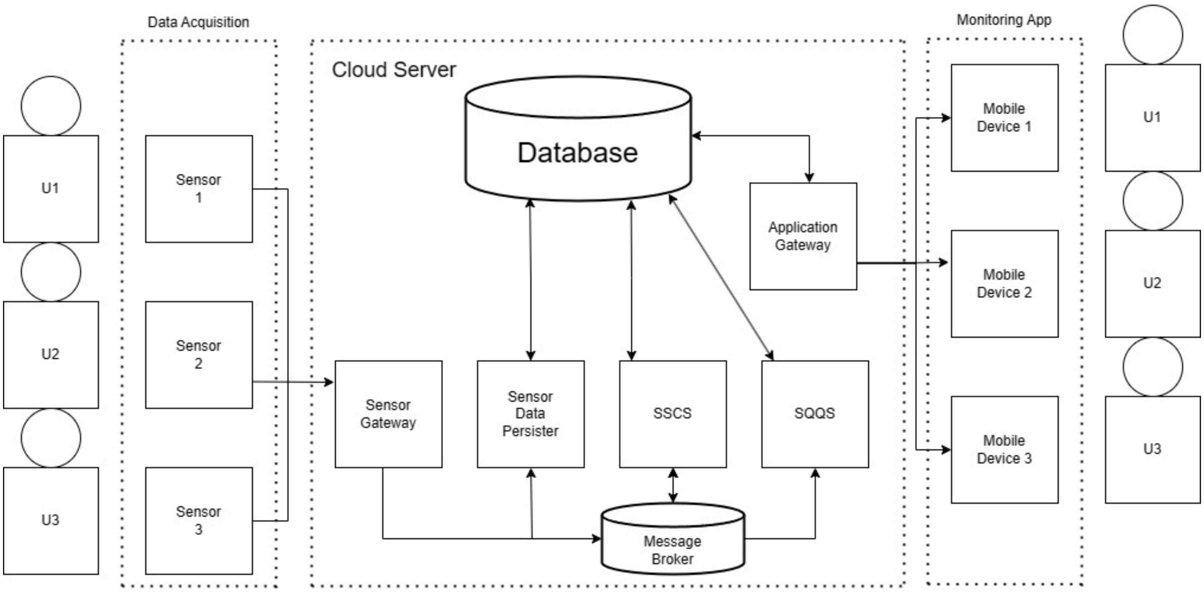
3. **Proposed Method.** The purpose of this study is to migrate a cloud-based sleep monitoring system from [9] with architecture shown in Figure 1 to become an edge-based sleep monitoring system. An edge device is used to implement the edge computing for the monitoring system. Several possible edge devices were compared in [23] which are Raspberry Pi, Jetson Nano, and Jetson TX2. In this study, Jetson Nano was used as the edge device considering that it has much lower cost than Jetson TX2 and higher performance than Raspberry Pi [23]. Jetson Nano that was used as the edge device had specifications as the following: 128-core Maxwell on GPU, Quad-core ARM A57 @ 1.43 GHz on CPU, 4 GB 64-bit LPDDR4 25.7 GB/s on RAM, 64 GB for storage, and TP Link TLWN725N 150 Mbps for Wi-Fi adapter dongle. The edge-based sleep monitoring system architecture is shown in Figure 2. Computing services and message broker that were inside cloud server are moved to inside of edge device. The cloud-based services and edge computing-based services have different requirements that need to be fulfilled.

Table 1. The summary of related work of the system architecture

Source	Proposed method	Result
[22]	Smart healthcare system using ensemble deep learning and feature fusion approaches	Creat a healthcare system that has increased accuracy for the prediction but with all of those privacy data being computed in cloud server, it is vulnerable from the attacks and it also relies on good network because it needs to be connected to the cloud server.
[20]	Real-time smart healthcare utilizing important measured values for critical care using Arduino UNO-based system and cloud computing	Creat a system that can measure values accurately and dashboard that can show the real-time result, but it lacks on security by computing privacy medical record on cloud server and also it needs good network environment to connect to the cloud server.
[19]	Internet of Medical Things (IoMT) that uses edge deep learning stack design with system architecture that initializes modules for each input that needs to be processed	Creat a system that achieved user data privacy, security, and low-latency by using edge-GPU architecture.
[21]	IoT-based student healthcare monitoring model that checks student vital signs and detects biological and behavioral changes via smart healthcare technologies	Creat a healthcare system that can predict student's condition with 99.1% accuracy by using Support Vector Machine (SVM). It also already implemented variety of security protocols by using third-party encryption, user authentication and credential mapping. However, this system relies on a good network environment because it needs to be connected to the cloud server to be able to do the computation.
[9]	IoT platform for sleep monitoring based on event-driven and microservice architecture	• Increase throughput by 92.59% and decrease response time 75.48% compared to monolith architecture. • Increase throughput by 34.76% and decrease response time 55.85% compared to microservices without event-driven architecture. • 7.81% and 17.3% slower for data processing from raw ECG data into sleep quality data.

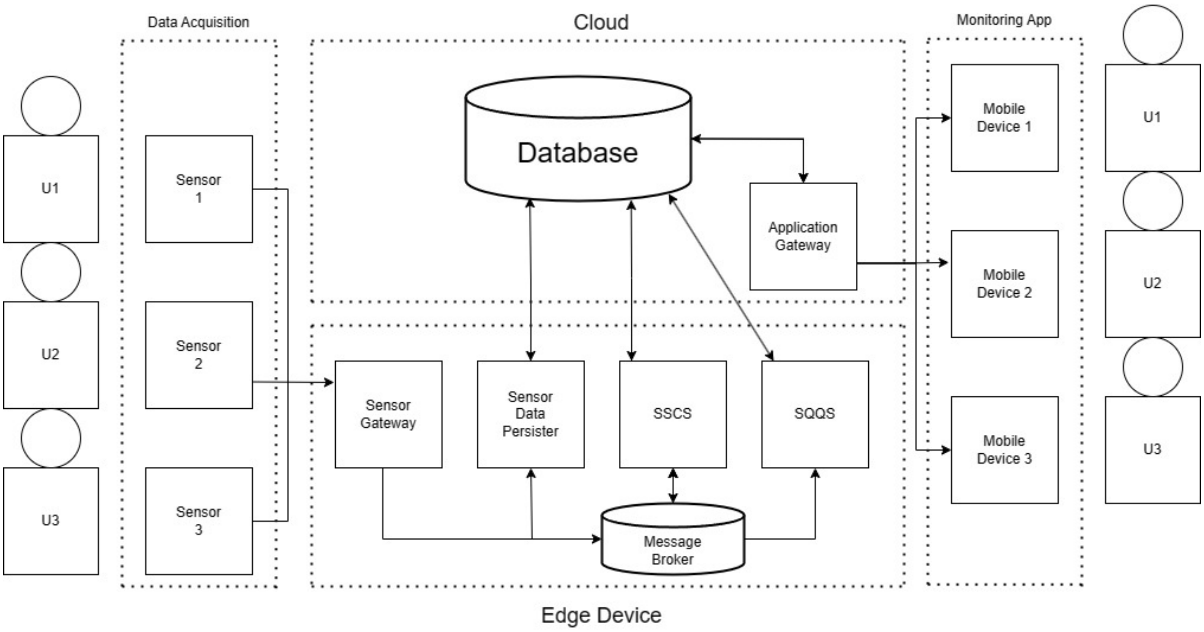
To migrate the cloud-based services into edge computing-based services so it can run at the edge device, the following steps were taken (as shown in Figure 3).

1) Build the sleep monitoring system project. In this study .jar files were created for each service because we used Java as programming language.
2) Build the docker images of the services based on the image architecture that is compatible with the edge device so that it can be run inside the edge device's container. In this study, Jetson Nano was used as the edge device and Jetson Nano's Operating System (OS) is Linux [24]; therefore, arm64 image architecture was used because it is compatible with Jetson Nano [25].
3) Run arm64-based message broker inside the container of the edge device for the services' communication intermediary.
4) Run all docker images of the services inside the edge device's container.



SSCS: Sleep Stage Classification Service
SQQS: Sleep Quality Quantification Service

FIGURE 1. Existing method



SSCS: Sleep Stage Classification Service
SQQS: Sleep Quality Quantification Service

FIGURE 2. Proposed method

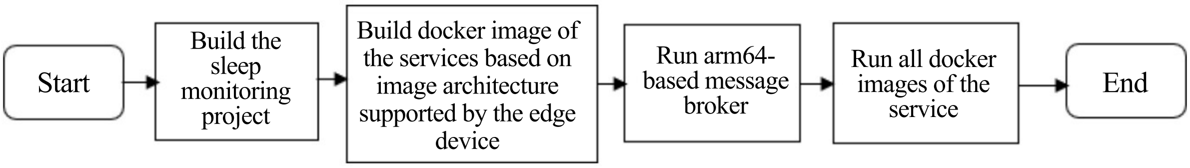


FIGURE 3. Services setup flowchart

Along with the changes of the sleep monitoring system architecture, the system computational processes flow also changed compared to the cloud-based sleep monitoring system. The goal is to perform the data processing on edge devices and only send the essential information which is the computational results to the database. A comparison of the two systems is illustrated by the process flowcharts presented in Figure 4 and Figure 5, for the cloud-based and edge-based systems, respectively. The proposed computational process flowchart of the system can be described as follows:

1) The sensor data are sent using Wi-Fi to the edge device (locally);
2) The services that run inside the edge device process the sensor data (data preprocessing, data analytics);
3) The computational results stored inside the database in cloud server using the Internet (globally).

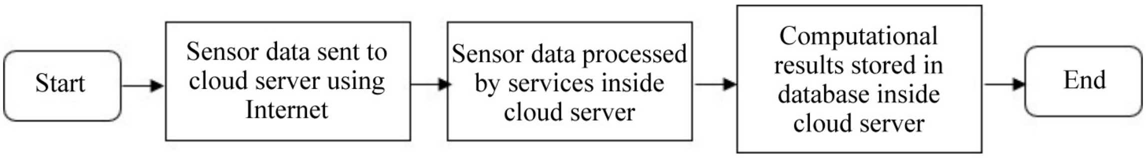


Figure 4. System computational processes by [9]

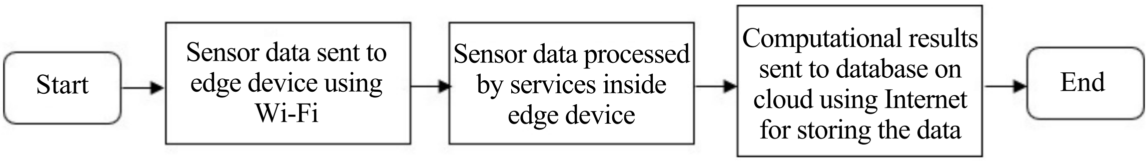


Figure 5. Proposed system computational processes

4. **Results and Discussion.** The evaluation of the performance of the proposed edge computing-based sleep monitoring system with microservices architecture was done by measuring three metrics which are the latency, response time, and throughput and then compare it with the cloud computing-based sleep monitoring system that run in google cloud with 2vCPU, 4GB RAM, and E-2 Medium type compute engine. The measurements were done using the JMeter [26] as follows.

4.1. Latency and response time.

4.1.1. *Evaluation scenario.* There is a slight difference between latency and response time; latency is measured from the time differences between just before sending the request and after the first response received. On the other hand, response time is measured from the time differences between just before sending the request and after all responses received. These two metrics can be evaluated in the same evaluation scenario. The system topology for the cloud system and edge system evaluation is shown in Figures 6 and 7, respectively. The evaluation scenario was done as follows.

1) Apache JMeter was used to simulate the sensors' request.
2) Assuming there was only one user for one edge device, one request was sent on each testing.
3) For cloud system, the request was sent using Internet (globally) and for edge system, the request was sent using Wi-Fi (locally).
4) The testing was done 50 times for each evaluation.

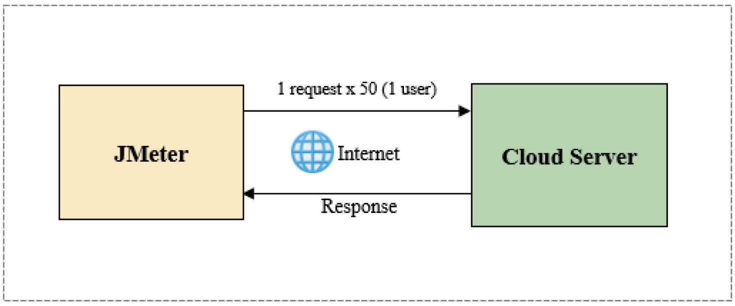


Figure 6. System topology on cloud for evaluation

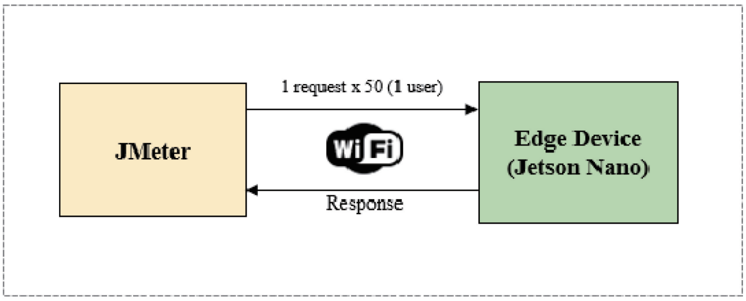


Figure 7. System topology on edge for evaluation

4.1.2. *Evaluation result.*

- Latency

The average latency for a request to reach the sensor and the first received response at the sensor was 43.2 milliseconds for cloud system and 31.52 milliseconds for edge system. Maximum latency achieved at 50 milliseconds which occurred in 2 percent of total request for edge system; on the other hand, cloud system achieved 69 milliseconds maximum latency, and 6.26 and 58.18 for standard deviation for edge and cloud system respectively shown in Figure 8. Cloud system's standard deviation was high because of two anomaly latency occurred that has significant difference. The evaluation shows that the edge system has lower average on latency which means edge system was quicker

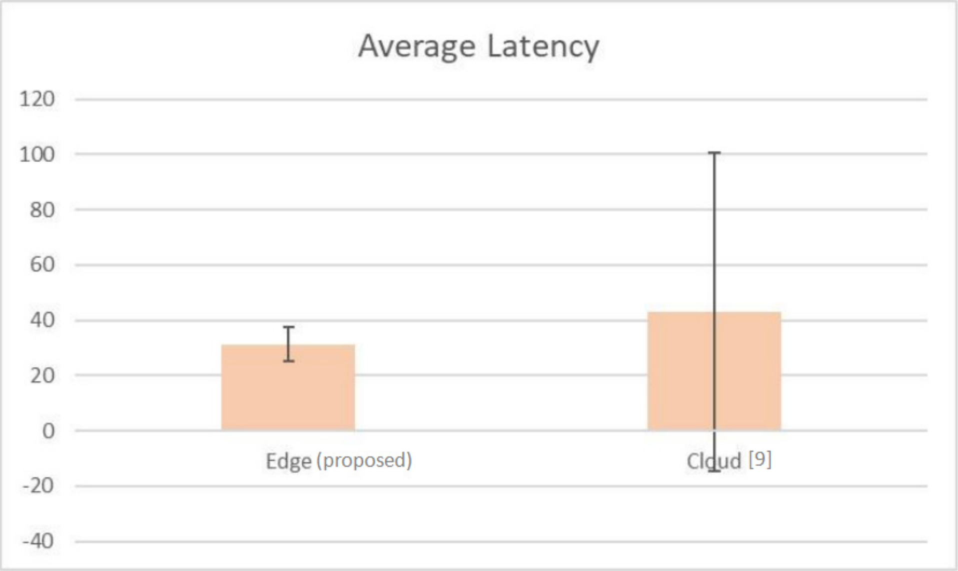


Figure 8. The average and standard deviation on latency

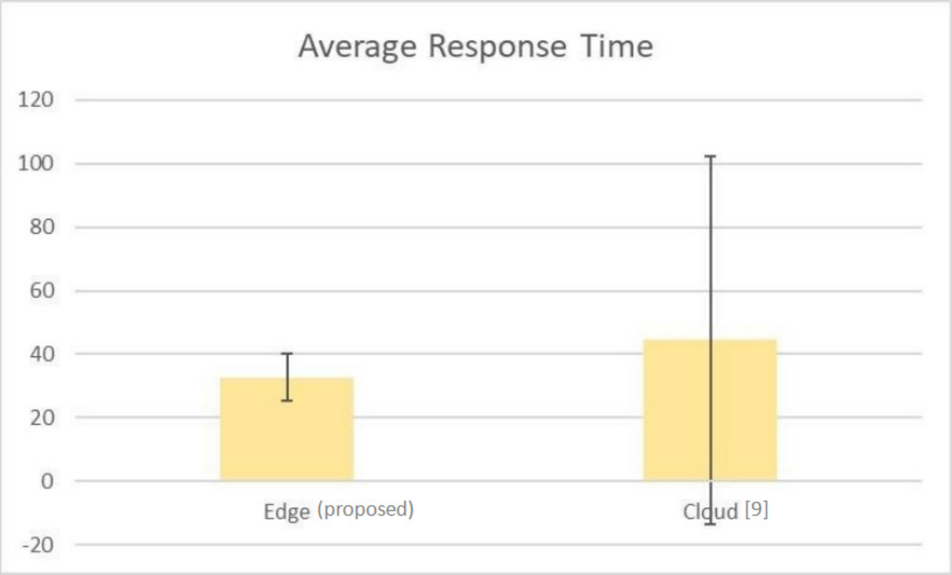


Figure 9. The average and standard deviation on response time

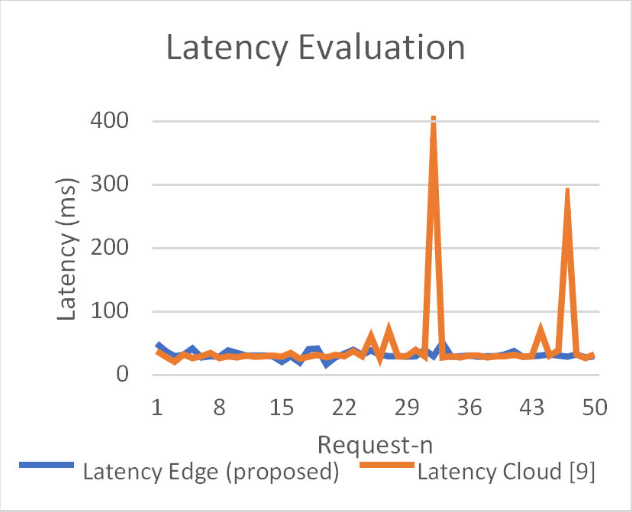


Figure 10. The evaluation of latency on 50 requests

on responding requests. The evaluation's result chart is shown in Figure 10. From the evaluation testing, we found that request sent cannot be above 50 because Jetson Nano cannot handle too many requests at a short time window.

- Response Time

The average response time for a request to reach the service in the edge system until receiving all the response in the sensor was 33.18 milliseconds and 44.4 milliseconds in cloud system. Maximum response time achieved at 59 milliseconds which occurred in less than 3 percent of total request on the other hand cloud system achieved 79 milliseconds maximum response time, and 7.65 and 58.32 for standard deviation for edge and cloud system respectively shown in Figure 9. Cloud system's standard deviation was high because of two anomaly response time occurred that has significant difference. In this result, the response time can be affected by the application load. It can be lower if the environment of the application is split between each service.

The evaluation results show that the edge system has a lower average response time, indicating that it is quicker at responding to requests. The response time evaluation is presented in Figure 11. However, we also observed that if the number of requests exceeds

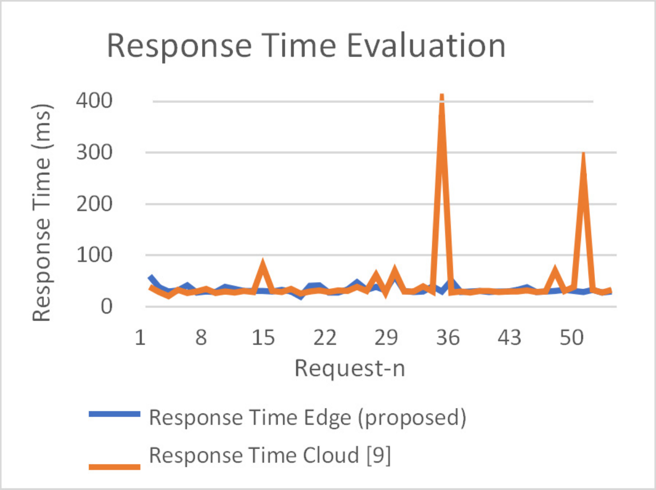


FIGURE 11. The evaluation of response time on 50 requests

50, the response time tends to increase due to the limited computing power of Jetson Nano.

4.2. **Throughput.**

4.2.1. *Evaluation scenario.* Throughput is measured from how many requests can be handled at a time. In this study, the requests number was converted in Kilobyte (KB) and second (s) was used as the unit of time. Figure 12 and Figure 13 show the system topology for throughput's evaluation scenario on cloud system and edge system, respectively. The evaluation scenario was done as follows.

1) Apache JMeter was used to simulate the sensors' request.
2) Assume there was 5 request that was sent from one user in one edge device.
3) For cloud system, the request was sent using Internet (globally) and for edge system, the request was sent using Wi-Fi (locally).
4) The testing was done 30 times for each evaluation.

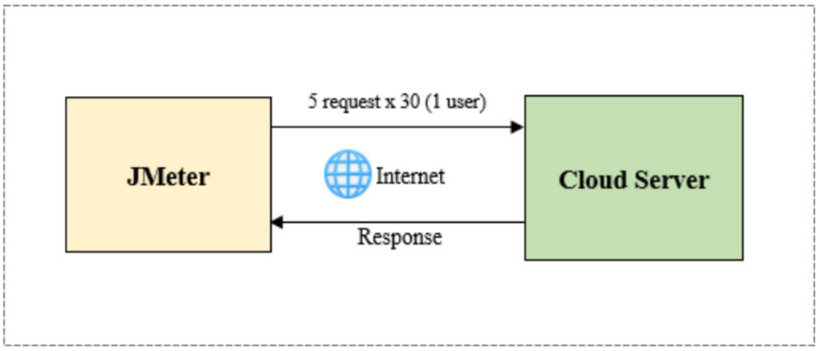


FIGURE 12. System topology on cloud for evaluation

4.2.2. *Evaluation result.* The average throughput for 5 requests at a time window was 25.2 KB/s for edge system and 29 KB/s for cloud system. Maximum throughput that had been achieved on edge system was 30.9 KB/s with minimum throughput of 10.8 KB/s; on the other hand, cloud system achieved 43.9 KB/s with minimum throughput of 4.3 KB/s. Each of these results was affected with the condition of high resource usage. The average and standard deviation chart is shown in Figure 14 and the evaluation chart is shown in Figure 15. The standard deviation for edge system was 4.44 and 8.69 for cloud system; it

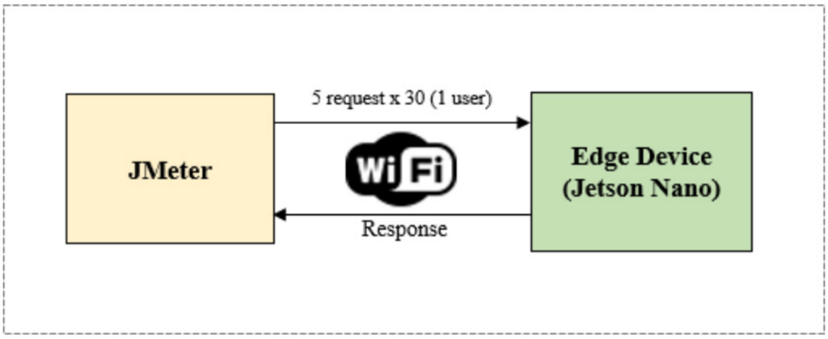


Figure 13. System topology on edge for evaluation

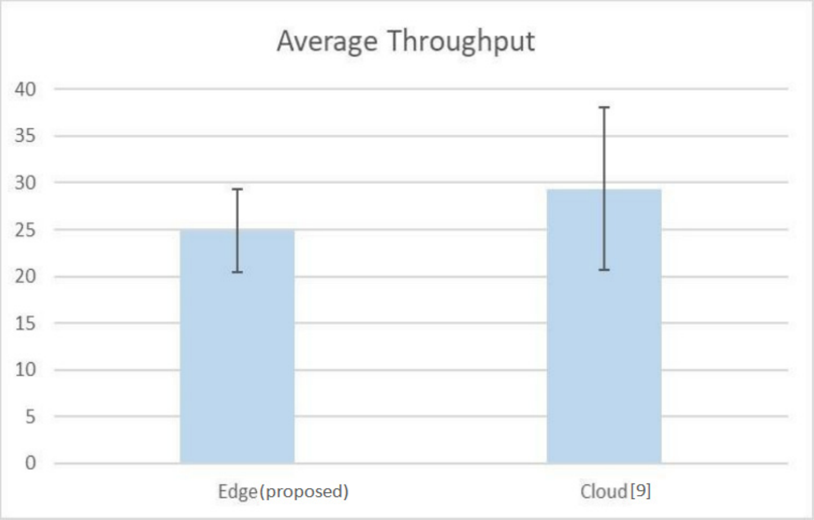


Figure 14. The average and standard deviation of throughput

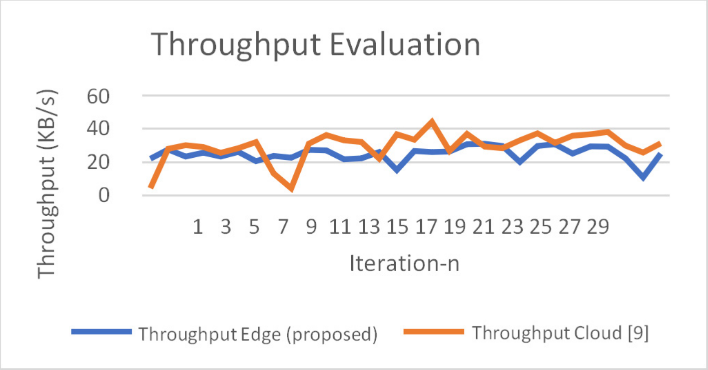


Figure 15. The evaluation of throughput on 30 iterations

indicates that the throughput on cloud system was more variative or unstable. From the evaluation testing we found that if the request was above 5, the container may encounter errors and shut down if it receives too many requests within a short time window, as the edge computing power may not be sufficient to handle the load.

5. **Conclusions.** This study provides a study for edge computing-based IoT architecture for sleep monitoring system by migrating a cloud computing sleep monitoring to the edge using Jetson Nano to check if the edge devices are able to compute near the source of the data. The system was built based on microservices architecture which made the

services can be deployed separately, and the system was also built using edge computing concept where the services were deployed inside of a container that was inside of an edge device with compatible architecture which is the arm64 architecture. The result data shows that the proposed edge computing-based sleep monitoring system with microservices architecture has lower latency and response time than the cloud computing-based sleep monitoring system designed by [9], but because of the limitation on the edge device's resource the throughput is lower than the cloud computing-based system. Therefore, the proposed system can be a solution for real-time sleep monitoring applications that require low latency and fast response time. After analyzing the performance and metrics data, it was found that combining edge computing with cloud computing can provide better results for all metrics. There are potential areas for improvement in this study, and further exploration of the potential of cloud edge computing is encouraged.

Acknowledgment. This publication is fully supported by Bina Nusantara University.

REFERENCES

[1] J. Bircher, Towards a dynamic definition of health and disease, *Medicine, Health Care and Philosophy*, vol.8, no.3, pp.335-341, 2005.

[2] M. Huber, J. A. Knottnerus, L. Green, H. Van Der Horst, R. A. Jadad, D. Kromhout, H. Smid et al., How should we define health?, *BMJ*, vol.434, DOI: 10.1136/bmj.d4163, 2011.

[3] F. Leonardi, The definition of health: Towards new perspectives, *International Journal of Health Services*, vol.48, no.4, pp.735-748, 2018.

[4] World Health Organization, *Basic Documents*, 2020.

[5] S. Dunlop, K. Swales and K. Trebeck, *Oxfam Humankind Index: The New Measure of Scotland's prosperity*, Oxfam GB, 2012.

[6] A. Sen, Development as freedom (1999), in *The Globalization and Development Reader: Perspectives on Development and Global Change*, J. T. Roberts, A. B. Hite and N. Chorev (eds.), Wiley, 2014.

[7] K. Ramar, R. K. Malhotra, K. A. Carden et al., Sleep is essential to health: An American Academy of Sleep Medicine position statement, *Journal of Clinical Sleep Medicine*, vol.17, no.10, pp.2115-2119, 2021.

[8] Consensus Conference Panel, N. F. Watson, M. S. Badr et al., Joint consensus statement of the American Academy of Sleep Medicine and Sleep Research Society on the recommended amount of sleep for a healthy adult: Methodology and discussion, *Sleep*, vol.38, no.8, pp.1161-1183, DOI: 10.5665/sleep.4886, 2015.

[9] N. Surantha, O. K. Utomo, E. M. Lionel, I. D. Gozali and S. M. Isa, Intelligent sleep monitoring system based on microservices and event-driven architecture, *IEEE Access*, vol.10, pp.42069-42080, 2022.

[10] N. Surantha, O. K. Utomo, M. I. Sani and B. Soewito, IoT system for sleep quality monitoring using ballistocardiography sensor, *International Journal of Advanced Computer Science and Applications*, vol.11, no.1, 2020.

[11] E. Simanjuntak and N. Surantha, Multiple time series database on microservice architecture for IoT-based sleep monitoring system, *Journal of Big Data*, vol.9, no.1, pp.1-19, 2022.

[12] K. Rajguru, P. Tarpe, V. Aswar, K. Bawane, S. Sorte and R. Agrawal, Design and implementation of IoT based sleep monitoring system for insomniac people, *2022 2nd International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pp.1215-1221, 2022.

[13] H. Ning, Y. Li, F. Shi and L. T. Yang, Heterogeneous edge computing open platforms and tools for Internet of Things, *Future Generation Computer Systems*, vol.106, pp.67-76, 2020.

[14] K. Chung and H. Yoo, Edge computing health model using P2P-based deep neural networks, *Peer-to-Peer Networking and Applications*, vol.13, no.2, pp.694-703, 2020.

[15] T. Lähderanta, T. Leppänen, L. Ruha et al., Edge computing server placement with capacitated location allocation, *Journal of Parallel and Distributed Computing*, vol.153, pp.130-149, 2021.

[16] H. Li, J. Zhang and L. Zhao, Vehicular high-definition maps cache based on dew computing, *2022 9th International Conference on Dependable Systems and Their Applications (DSA)*, pp.1067-1068, 2022.

[17] X. Zhang, Y. Liu, Z. Cao, H. Zhou and F. Zhang, An efficient service migration model based on improved genetic algorithm in mobile edge computing environment, *International Journal of Innovative Computing, Information and Control*, vol.17, no.4, pp.1401-1419, DOI: 10.24507/ijicic.17.04.1401, 2021.

[18] O. K. Utomo, N. Surantha, S. M. Isa and B. Soewito, Automatic sleep stage classification using weighted ELM and PSO on imbalanced data from single lead ECG, *Procedia Computer Science*, vol.157, pp.321-328, 2019.

[19] A. M. Rahman and S. M. Hossain, An Internet-of-Medical-Things-enabled edge computing framework for tackling COVID-19, *IEEE Internet of Things Journal*, vol.8, no.21, pp.15847-15854, 2021.

[20] M. M. Khan, S. Mehnaz, A. Shaha, M. Nayem and S. Bourouis, IoT-based smart health monitoring system for COVID-19 patients, *Computational and Mathematical Methods in Medicine*, 2021.

[21] A. Souri, Y. M. Ghafour, M. A. Ahmed, F. Safara, A. Yamini and M. Hoseyninezhad, A new machine learning-based healthcare monitoring model for student's condition diagnosis in Internet of Things environment, *Soft Computing*, vol.24, no.22, pp.17111-17121, 2020.

[22] F. Ali, S. El-Sappagh, R. S. Islam, D. Kwak, A. Ali, M. Imran and K.-S. Kwak, A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion, *Information Fusion*, vol.63, pp.208-222, 2020.

[23] A. A. Süzen, B. Duman and B. Şen, Benchmark analysis of Jetson TX2, Jetson Nano and Raspberry Pi using deep-CNN, *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp.1-5, 2020.

[24] A. Basulto-Lantsova, J. A. Padilla-Medina, F. J. Perez-Pinal and A. I. Barranco-Gutierrez, Performance comparative of OpenCV Template Matching method on Jetson TX2 and Jetson Nano developer kits, *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.812-816, 2020.

[25] I. Rasmussen, S. Kvalsvik, P. A. Andersen, T. N. Aune and D. Hagen, Development of a novel object detection system based on synthetic data generated from unreal game engine, *Applied Sciences*, vol.12, no.17, 8534, 2022.

[26] APACHE Software Foundation, *Apache JMeter Glossary*, http://jmeter.apache.org/usermanual/glossary.html, 2013.