Journal of Statistics & Management Systems
ISSN 0972-0510 (Print), ISSN 2169-0014 (Online)
Vol. 27 (2024), No. 3, pp. 721–732
DOI : 10.47974/JSMS-1281

Mathematical framework for real-time data processing in edge computing : Context-aware priority scheduling analysis

V. Dankan Gowda *
Department of Electronics and Communication Engineering
BMS Institute of Technology and Management
Bangalore
Karnataka
India

V. Nuthan Prasad †
Department of Electronics and Communication Engineering
M S Ramaiah Institute of Technology
MSR Nagar, MSRIT Post
Bengaluru
Karnataka
India

K. D. V. Prasad §
Department of Research
Symbiosis Institute of Business Management
Hyderabad
Telangana
India

and

Department of Research
Symbiosis International (Deemed University)
Pune
Maharashtra
India

* *E-mail:* dankan.v@bmsit.in (Corresponding author)
† *E-mail:* nutan2u@gmail.com
§ *E-mail:* kdv.prasad@sibmhyd.edu.in

Kottala Sri Yogi ‡
Department of Operations
Symbiosis Institute of Business Management
Hyderabad
Telangana
India

and

Department of Operations
Symbiosis International (Deemed University)
Pune
Maharashtra
India

Manojkumar Shivalli Boraiah @
Department of Electronics & Communication Engineering
BGS Institute of Technology/ACU
B G Nagara, Mandya
Karnataka
India

Mirzanur Rahman #
Department of Information Technology
Gauhati University
Assam
India

Abstract

The increased demand for real-time data processing has brought into the emergence of edge computing as an important paradigm that helps to fulfil all such requirements on time-sensitive applications as they have Latency and efficiency prerequisites. The work offers a new framework designed to conduct real-time processing of data, and it relates to situations when the data is located at the network edge. To achieve the best scheduling and edge processing efficiency, this work proposed a "Context-Aware Priority Scheduling Framework" that utilizes contextual information. Choosing the ideal processing method in this architecture requires that one understands the context with which to undertake data collection. The framework has the ability to automatically prioritize data processing activities based on various variables, which include location residence, network situation, device capabilities and web needs or preferences by data consumers. The overall responsiveness of a system is improved and becomes more effective when the critical tasks are given high

‡ *E-mail:* **Sriyogi.k@sibmhyd.edu.in**
@ *E-mail:* **manojkumarsb@bgsit.ac.in**
# *E-mail:* **mr@gauhati.ac.in**

priority. When two scheduling approaches context-aware and priority are united, they not only enhance the efficiency of edge computing, but also provide with a systematic basis for intelligent decision-making in the edge environment. The given framework is able to mitigate the latencies of data processing while fully utilizing available resources with a series of experiments and performance evaluations.

Subject Classification: *94A60, 20C05, 20C07.*

Keywords: *Real-time data processing, Edge computing, Context-awareness, Priority scheduling, Framework, Latency, Efficiency and contextual information.*

1. Introduction

The modern domain has been characterized by an increased demand for efficient and low latency processing solutions, a trend mainly triggered by the escalation in real-time data driven applications. Due to the challenges posed by these applications, edge computing has emerged as a transformative concept of bringing data processing closer to the source of data. The current paper is A Study of Real-Time Data Processing in Edge Computing that aims to enhance the efficiency and response time in edge computing environments. The purpose of Smart Environments is just to make people's lives better in a general way. This study delves into the area of smart computing environments, with a particular focus on the integration of IoT. The IoT represents a new concept pivotal for smart environment generation. This can be attained by leveraging the capabilities of cloud computing coupled with smart sensing devices that already exist in IoT gadgets [1,2]. Numerous research works have indicated an increased trend of the IoT in terms of devices emerging, hence another disposal immense pool. In this study, An Edge computing based on Context Awareness for Priority scheduling was successfully implemented as desired via Interdisciplinary approach om [3, 4]. The research will concentrate on the design, development and evaluation of a just-in-time framework by approaching it through both theoretical analysis and practical experimentation. The performance is evaluated by comparing the processing time, resource utilization and responsiveness with those of normal scheduling techniques.

2. Literature Survey

In this literature review the areas that have been considered include; edge computing, real-time data processing, context awareness as well as priority scheduling [5]. In this regard, discuss the background for our study "fidelity" by describing the current situation and determining existing issues as well as articulating those knowledge voids that will be fulfilled with our research [6,7]. The idea of "edge computing" has developed recently as a response to the issues triggered by centralization latency and wastefulness. Researchers have presented several architectures [8,9], different deployment models and optimization strategies for edge settings. This all-embracing network makes the process of collecting, analyzing and employing environmental data easier for users. The IoT is still small but, combined with other technologies to meet different requirements, we can create an intelligent setting that will enhance people's lives and the world as a whole[10,11]. The data collected from IoT devices is the backbone of a smart environment and is used in conjunction with complex decision-making algorithms run on the cloud, which have access to extensive computing resources. Integrating Internet of Things (IoT) devices with sensors to gather environmental data is, according to [12,13], a key characteristic of a Smart Environment. Some examples of IoT-based smart landscapes are depicted graphically in Figure 1. Kevin Ashton first used the phrase "Internet of Things" (IoT) in 1999 in reference to logistics and supply chain management [14]. Through the use of embedded sensors, IoT connects physical objects to the web, allowing electronic gadgets to gather data about their environment and respond accordingly.

Ashton's demonstration of this notion was the use of RFID tags on objects in the supply chain, which, when linked to the Internet, allowed for the automated verification of the availability of commodities[15,16]. This information sharing allows for hands-free operation of these devices from a distance. The "Things," communication technology, and computing aspects form the trinity upon which the Internet of Things rests. As shown in Figure 2, these elements work together to bring IoT's possibilities to life. Services that are dependable and of high quality require real-time

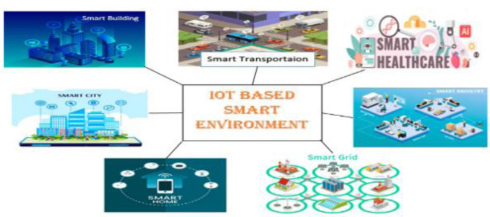

Figure 1

Connected Smart Environment

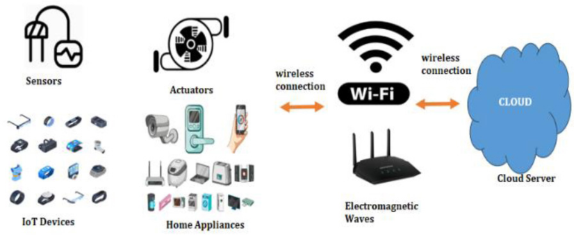

Figure 2

IoT components

processing of data collected from smart surroundings[17,18]. Data can be computed and stored conveniently close to its original source, at the network's edge, thanks to edge computing[19]. This method guarantees the delivery of services in real time and efficiently addresses latency issues.

There is a significant knowledge gap when it comes to integrating edge computing, real-time processing, context awareness, and priority scheduling, despite the fact that extant literature provides insights into these ideas individually[20-23]. The difficulty of setting priorities for data processing activities in edge contexts while taking environmental considerations into account has been the subject of relatively few studies. To bridge this gap and maximize real-time data processing in edge computing, a holistic framework that integrates these areas is required.

3. Proposed Scheduling Algorithms

Figure 3 provides a visual representation of the scheduling system. In this setup, sensors built into IoT devices continuously observe and record data about their surrounding environment. The data collected by the sensors is then sent to the Edge Device using various methods of transmission. When a predetermined event occurs, the associated sensor

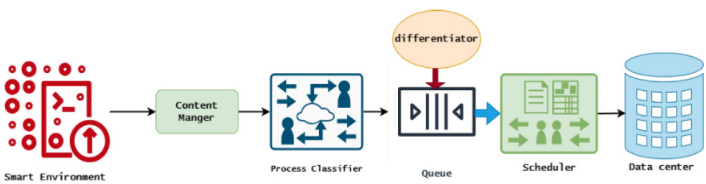

Figure 3

Proposed Edge Computing System

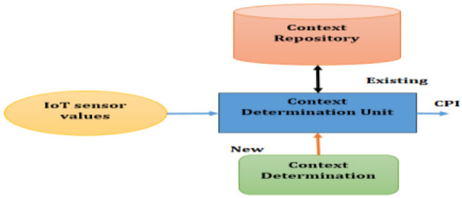

Figure 4

Context Management Process

in the IoT device sends the appropriate data to the Edge or Cloud for analysis.

The appropriate next step for the IoT device to take is then established. We call this IoT-enabled, smart-environment-orchestrated procedure simply a "IoT process." The Cloud server can run a large number of IoT activities, while the Edge server has limited resources and cannot do so. Processes are prioritized and queued based on these priorities before being run either locally at the Edge or remotely in the Cloud. With the help of sensors, a large number of IoT devices can contribute to environmental monitoring within the context of a Smart Environment. When it comes to picking up on the world around you, these sensors are indispensable. Figure 4 is a graphical representation of the two main parts of the contextual management unit: the Context Repository and the Context Determination Unit. The criticality of a process can be estimated using this CPI number. However, the context model is triggered if the sensor data is unique. This adaptability guarantees that each user's tastes and wants will be taken into account while differentiating. If two processes have the same CPI value, the Differentiator module will create a "Second Priority Index" (SPI). This allows for efficient differentiation of processes with similar CPI values. The calculated SPI scores are then ranked from highest to lowest. According to the established order of importance, this structure determines whether a given process will be handled at the Edge or in the Cloud.

4. Queue Manager

Figure 5 depicts the multilayer queue scheduling strategy used by the scheduling algorithm. The system classifies tasks into three distinct categories based on their importance: High Priority tasks, which are mission-critical, Medium Priority tasks, which are delay-sensitive, and Low Priority tasks, which can tolerate delays. In accordance with their

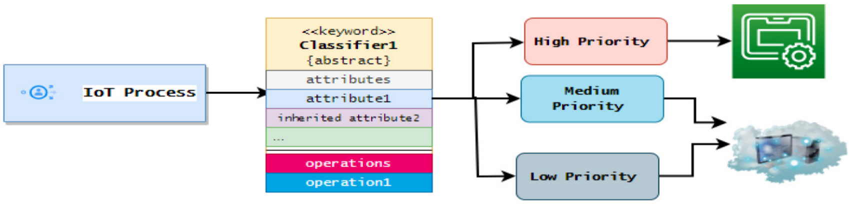

Figure 5

Proposed Queue Manager

relative importance, these tasks are placed in the appropriate priority queues.

High, medium, and low criticality levels are used to categorize the IoT processes. However, there are situations in which two processes within the same criticality level have the same criticality value (CPI). Therefore, it is essential to differentiate processes operating at the same criticality level. These issues have prompted the development of the following methods for distinguishing between similar processes:

Proposed Priority Algorithm: This algorithm specifies a procedure to be followed in order to distinguish amongst Internet of Things operations with the same criticality level.

Step 1: Identify processes within the same criticality level with identical CPI values. ***Step 2:*** Utilize the Second Priority Index (SPI) as generated by the Differentiator module to distinguish between these processes.

Step 3: Sort processes based on ascending SPI values.

Step 4: Allocate processes for execution at either the Edge or the Cloud in accordance with their SPI-assigned sequence.

Processes are ranked from most critical to least critical using this algorithm. The Edge is the location where mission-critical tasks run to execute while the Cloud is exactly where less intervention responsibilities operate. To do so, every priority queue performs the tasks in the order they were inserted into the queue called as First-Come-First-Serve (FCFS). The use of these algorithms makes it possible to deal with the difficulty in distinguishing between processes of equal criticality within a level effectively.

5. Results and Discussion

The previous section outlined the procedures that should be put in place to assess how successful were a build of algorithms. This process includes testing these algorithms to know how efficient they are, their accuracy and whether or not they suffices the job to be done. In this analysis let us find out if the scheduling algorithm achieves what it sets to achieve. The performance of the developed algorithms are evaluated using a simulation method. Simulating something refers to the process in which a model is developed within a computer to reflect some feature of the real world. This virtual setting is applied launch the algorithms and observe their actions and outcomes. The role of the simulator is to evaluate the performance of the suggested scheduling algorithm as compared to other algorithms in its class. The process of evaluation is in comparing the effectiveness of the proposed scheduling algorithm with other algorithms that are already being used. To measure the efficiency of a designed algorithm, an output from simulation is quantified and analyzed with respect to predetermined criteria. We make use of performance indicators such as response time, execution time, resource utilization and etcetera to evaluate the efficiency of an algorithm. For instance, the suggested one over Existing Shortest Job First (E-SJF) is Priority-based shortest job first(P-SJF). Figure 6 contrasts the average Turnaround Time (TAT), Cost, and Energy usage of IoT procedures using the P-SJF algorithm vs the E-SJF method. The chart shows the percentage gains made by the new algorithm compared to the old one for each of these metrics. When compared to the suggested P-SJF algorithm, the E-SJF method shows an average TAT of 411.24 ms, whereas the P-SJF algorithm has an average TAT of 52.37 ms. With a decrease in TAT of 358.87 ms, the effectiveness of the suggested method is demonstrated, allowing for more rapid decision-making in highly time-sensitive procedures. The current E-SJF algorithm has a maximum TAT of 425.08 ms, making it unsuitable for use in highly time-sensitive processes. When compared to the HCP_TAT ratio, the P-SJF algorithm's TAT of 16.15 ms for the same high critical process is a significant performance improvement of 96.20%. Furthermore, the P-SJF algorithm improves the average TAT by 87.26 percent when compared to the standard SJF algorithm. The effective execution of IoT activities at the Edge is responsible for this enhancement, since it guarantees the shortest possible execution durations within each critical level. When compared to the standard SJF algorithm, the P-SJF algorithm also results in a notable reduction of energy consumption, by 91.09%. In addition to optimizing efficiency, the suggested P-SJF algorithm minimizes expenses by 73.19 percent when compared to the standard SJF method.

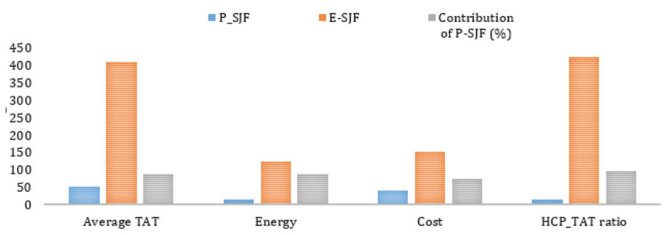

Figure 6

Performance of Proposed Algorithm

Optimal process execution in IoT devices requires careful attention to energy consumption. How much juice is left in an IoT device once a process has finished running has a direct bearing on how much else it can do. When many operations have the same importance, the IoT device's remaining power can be used to determine the order in which they will be run. This paper contrasts the Priority-based Energy (P-Energy) method, which takes priority and remaining energy into account, with the current approach, which does not. Current algorithm is called E-Energy.

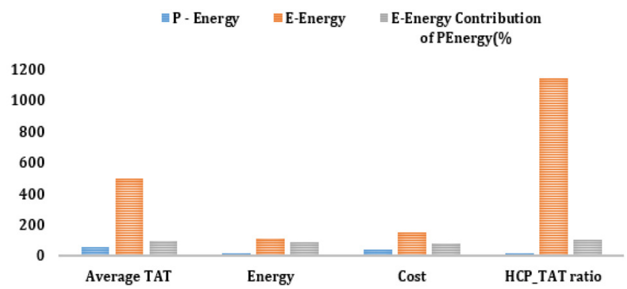

Figure 7

Performance of P-Energy algorithm

In Figure 7, we can see how the proposed P-Energy method stacks up against the standard E-Energy approach. Average Turnaround Time (TAT), cost, and energy consumption within the EDGE_CLOUD framework are compared. The priority-based P-Energy algorithm shows an average TAT of 54.81 ms when making scheduling decisions that take into account the remaining energy level of IoT devices. In contrast, the current E-Energy algorithm takes an average of 494.52ms to complete its execution. The suggested method's benefits are highlighted by the significant difference in execution times between the two algorithms (439.71 ms). The P-Energy algorithm makes actuators actuate in IoT devices by speeding the process with a factor of 439.71ms in the Smart Environment. To do that, the IoTPriority of the IoTprocess and how much energy is left in the IoTdevice are considered at once. Based on process priority and the remaining energy level indicated above, it is clear that the suggested P-Energy algorithm implements efficient IoT process scheduling optimally. The above highlighted method ensures faster actuation of Smart Environment ad at the same time improve on how it was effective and efficient in its operations.

6. Conclusion

The focus on this research is meant to demonstrate how critical it is to resolve challenges related to latency-sensitive applications, and our framework offers a new model which merges the issues encompassing real-time data processing, context awareness, and priority scheduling. A comparison has been made based on key measurements including average Turnaround Time (TAT), cost, and energy consumption when contrasted to those algorithms used presently. The objective in this analysis is to show that the algorithms proposed are better as compared to the methods used today. The analysis of the acquired data reveals that suggested algorithms work well and fast. Using these benchmarks, the proposed algorithms have proven to be more effective than state-of-the-art methods. It is noted that, when comparing the proposed algorithms to the well known standards, they have shown better performance and may serve as a hope for further enhancing some features related to the target systems. This empirical verification provides more support for the validity and practicality of the suggested methods.

References

[1] M. Hosseinzadeh, “Task Scheduling Mechanisms for Fog Computing: A Systematic Survey,” in *IEEE Access*, vol. 11, pp. 50994-51017 (2023).

[2] C. Barros and O. Teixeira, “Proposal of a Context-aware Task Scheduling Algorithm for the Fog Paradigm,” 2022 Seventh International Conference on Fog and Mobile Edge Computing (FMEC), Paris, France, pp. 1-8 (2022).

[3] J. X. Liao and X. W. Wu, "Resource Allocation and Task Scheduling Scheme in Priority-Based Hierarchical Edge Computing System," 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Xuzhou, China, pp. 46-49 (2020).

[4] P. Paymard and M. Orooji, "Task Scheduling Based on Priority and Resource Allocation in Multi-User Multi-Task Mobile Edge Computing System," 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, pp. 1-7 (2019).

[5] S. Reshma Sultana and K. Abinaya, "Priority based Scheduling Algorithm using Divisible Load Theory in Cloud," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, pp. 771-776 (2020).

[6] S. Lipsa and K. Cengiz, "Task Scheduling in Cloud Computing: A Priority-Based Heuristic Approach," in *IEEE Access*, vol. 11, pp. 27111-27126 (2023).

[7] W. Guo and W. Liu, "A Task Priority-based Resource Scheduling Algorithm for Container-based Clouds," 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), Chongqing, China, pp. 268-273 (2021).

[8] G. Kaur and S. Midha, "A Preemptive Priority Based Job Scheduling Algorithm in Green Cloud Computing," *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Noida, India, pp. 152-156 (2016).

[9] M. Laroui Afifi and M. Guizani, "Edge and fog computing for IoT: A survey on current research activities & future directions", *Comput. Commun.*, vol. 180, pp. 210-231, Dec. (2021).

[10] M. R. Alizadeh and E. Akbari, "Task scheduling approaches in fog computing: A systematic review", *Int. J. Commun. Syst.*, vol. 33, no. 16, pp. e4583, Aug. (2020).

[11] N. Kaur and R. Kumar, "A systematic review on task scheduling in fog computing: Taxonomy tools challenges and future directions", *Concurrency Comput. Pract. Exper.*, vol. 33, no. 21, pp. e6432, Nov. (2021).

[12] C. Barros and H. Paredes, "Survey on Job Scheduling in Cloud-Fog Architecture", 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1-7, (2020).

[13] Tang Ling and Shibo He, "Multi-user computation offloading in mobile edge computing: A behavioral perspective", *IEEE Network*, vol. 32.1, pp. 48-53 (2018).

[14] Santosh Kumar & Gupta, Anand Kumar. A novel approach of unsupervised feature selection using iterative shrinking and expansion algorithm, *Journal of Interdisciplinary Mathematics*, 26:3, 519-530 (2023), DOI: 10.47974/JIM-1678.

[15] Naziya & Chinamuttevi, Veera Sivakumar. A novel RF-SMOTE model to enhance the definite apprehensions for IoT security attacks, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:3, 861-873 (2023), DOI: 10.47974/JDMSC-1766.

[16] Selvakumar & Chaturvedi, Abhay. Securing networked image transmission using public-key cryptography and identity authentication, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:3, 779-791 (2023), DOI: 10.47974/JDMSC-1754.

[17] H. G. Govardhana Reddy & K. Raghavendra. Vector space modelling-based intelligent binary image encryption for secure communication, *Journal of Discrete Mathematical Sciences and Cryptography*, 25:4, 1157-1171 (2022), DOI: 10.1080/09720529.2022.2075090.

[18] Yuan, C & Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. *J*, *2*, 226-235 (2019).

[19] Sulaiman & Sulaiman, Ranen M.A new secant method for minima one variable problems, *Journal of Interdisciplinary Mathematics*, 26:6, 1015–1021 (2023), DOI: 10.47974/JIM-1601.

[20] Xu-Zheng CHANG and Wen-Bin JIAO, Improvement and Implementation of Kubernetes Resource Scheduling Algorithm., vol. 29, no. 07, pp. 256-259 (2020).

[21] Kumar, Anil , Gupta, Amit Kumar , Panwar, Deepak , Chaurasia, Sandeep & Goyal, Dinesh. Operating system security with discrete mathematical structure for secure round robin scheduling method with intelligent time quantum, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1519–1533 (2023).

[22] Kushwaha, Satpal Singh, Joshi, Sandeep & Gupta, Amit Kumar. An efficient approach to secure smart contract of Ethereum blockchain using hybrid security analysis approach, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1499–1517 (2023), DOI: https://doi.org/10.47974/JDMSC-1815.

[23] Joshi, Ruchi, Mathur, Priya, Gupta, Amit Kumar, Singh, Suyesha, Paliwal, Vismita & Nayar, Sejal. Mathematical modeling of intelligent system for predicting effectiveness of premenstrual syndrome, *Journal of Interdisciplinary Mathematics*, 26:3, 551-562 (2023).