Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12682 p-ISSN : 2541-044X

Performance Analysis of Scheduling Algorithms on Fog Computing using YAFS

Dimas Nurcahya[1)*], Siti Amatullah Karimah[2)], Satria Akbar Mugitama[3)]
[1)2)3)] Informatics, School of Computing, Telkom University, Bandung, Indonesia
[1)] dimasnurcahya@student.telkomuniversity.ac.id, [2)] karimahsiti@telkomuniversity.ac.id,
[3)] satriamugitama@telkomuniveristy.ac.id

Submitted : Jul 11, 2023 | **Accepted** : Jul 13, 2023 | **Published** : Jul 13, 2023

Abstract: A smart device that has seen more development is the Internet of Things (IoT). An IoT system implementation requires a device that can accept and handle various sorts of data. Fog Computing is a solution to the issue since the IoT demands a device that can provide Real-Time. Certainly, load balancing involves scheduling the IoT devices and data used. Because Cloud and Fog Computing models enable data growth management and deployment planning, which necessitate a quicker response from platforms and applications, processing power scheduling is essential. The purpose of this study is to evaluate the performance of effective scheduling algorithms that adhere to these computing models platform requirements. The scheduling algorithm that can produce the lowest Processing Time and the resulting Time Efficiency is more efficient can be called the best scheduling algorithm. In this research, the author analyzes the performance of scheduling algorithms in the form of Round Robin and Priority Scheduling on Fog Computing. In this research, testing was carried out by creating a scenario of the effect of increasing the number of Fog Nodes and Devices used. The average result of scenario testing obtained for processing time for Round Robin is lower, and the highest Time Efficiency for Round Robin over Priority Scheduling is 11%. With these test results, the Round Robin scheduling algorithm has a simpler level of complexity. So, it can be concluded that Round Robin belongs to the category of the best scheduling algorithm in this case.

Keywords: Fog Computing; Round Robin; Priority Scheduling; Processing Time; Time Efficiency

INTRODUCTION

Cloud computing (CC) is defined by NIST (National Institute of Standards and Technology) as a model for providing universal and on-demand network access to a pool of reconfigurable computing resources, such as networks, servers, storage, applications, and services, that can be quickly provisioned and released with little management work or service provider interaction (de Donno et al., 2019). However, Cloud Computing has shortcomings in data transfer due to infrastructure limitations and networks that greatly hamper performance. Therefore, there is a new paradigm called Fog Computing (FC), which acts as a Middleware between the cloud and IoT (Abdali et al., 2021).

Fog Computing is a new architecture that acts as a layer between the two to create a bridge between the Cloud and the Internet of Things (IoT) world and provide services directly to the Network Edge (Margariti et al., 2020).

The establishment of Fog computing easily facilitates work between Cloud centers and devices residing at the Network edge and thus turns out to be a better solution to overcome the problems presented by Cloud Computing (Neware, 2019a).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12682 p-ISSN : 2541-044X

Cisco has researched that in 2019 and 2020, many devices will be connected to the Internet, including data generated by Brainware. Data generated by the IoT is then interpreted, processed, analyzed, and stored at the Network edge (Neware, 2019b). As the number of devices grows, Fog Computing has become a paradigm that is poised to optimize various key Quality of Service (QoS) requirements such as Latency, Bandwidth limitation, response time, scalability, privacy, and security (Singh et al., 2021). With so much load, it can affect the performance of Fog Computing. Therefore, a scheduler is needed to manage the given load.

Simulation and performance testing of scheduling algorithms are done using YAFS tools. Yet Another Fog Simulator (YAFS) is a Fog Computing simulator tool designed to analyze application design and incorporate strategies for placement, scheduling, and routing (Lera et al., 2019a). YAFS uses a generic library for discrete event simulation scenario generation called Simpy. Simpy is a powerful and stable discrete-event simulator implementation that contains functions for the definition of processes (active components) and shared resources (such as network links and queues). With this, it is possible to execute simulations in three modes: as fast as possible, in real time, or manually to step through events.

In this research, the author analyzes the performance of the Scheduling algorithm on Fog Computing using YAFS Tools. The scheduling algorithms used are Round Robin scheduling and Priority Scheduling. This algorithm is one of the most important aspects of the search process that must be handled well to run well (Sakshi et al., 2022). Scheduling algorithms are needed to guide the process successfully (Jin & Yu, 2022b). Fast resource allocation is essential in fog computing, as tight response times are required for many applications that use fog computing (Choudhari et al., 2018a). Scheduling can enable various types of systems to improve performance efficiency with low resource costs and a simple architecture. Round Robin distributes jobs for processing at each level of the architecture (Guevara & da Fonseca, 2021). Meanwhile, Priority Scheduling minimizes the time spent by service requests in the system queue and helps achieve high performance by making efficient resource provisioning (Choudhari et al., 2018b).

LITERATURE REVIEW

Fog Computing

The Internet of Things (IoT) supports billions of physical devices for data collection and transmission to various administrations, such as environmental monitoring, infrastructure planning, and home automation (Abdali et al., 2021). So far, Cloud Computing has proven to be a capable way to process and store such data. However, it turns out that the use of Cloud Computing alone cannot solve unusual problems such as the increasing demand for time- and speed-sensitive applications and the limited transmission speed of the system (Neware, 2019a). Fog Computing has superior services compared to Cloud Computing, thanks to limited information storage and delivery for end users who do not have a central Cloud (Abdali et al., 2021).

The establishment of Fog computing easily facilitates the work between the Cloud center and the devices located at the Network Edge, thereby turning out to be a better solution to overcome the problems presented by Cloud computing (Neware, 2019b). For example, time-sensitive requests such as Augmented Reality, Audiovisual Streaming, and Video Game play cannot be supported by the Cloud. Therefore, these issues are addressed by Fog Computing (Sabireen & Neelanarayanan, 2021).

Because Cloud Computing has several shortcomings, such as data transmission, being time sensitive, and being very far from the device, this weakness affects some devices that require speed in data transfer and for which the distance between devices is not too great. So, there is a paradigm called Fog Computing to overcome the problems that Cloud Computing has. With so many devices connected to Fog Computing, Scheduling algorithms are needed to simplify the process of working with fog computing. An effective scheduling algorithm not only minimizes resource consumption but also ensures that new jobs can be completed as quickly as possible (Murad et al., 2022).

Round Robin Algorithm

Sarayut Phorncharoen and Worawat SA-Ngiamvibool (2018) conducted tests using the Scheduling algorithm, namely, Round Robin. Round Robin (RR) is the most common Load Balancing strategy

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12682 p-ISSN : 2541-044X

supported by major Cloud providers (Tychalas & Karatza, 2020). The algorithm used to facilitate the performance of Fog Computing in dividing the load, commonly referred to as load balancing, the results obtained using the Round Robin Scheduling algorithm can reduce 22.97% AWT (Average Waiting Time), 22.13% ATT (Average Turnaround Time), and 30.26% NCS (Number Context Switch) for a data set of 50 processes (Phorncharoen & Sa-Ngiamvibool, 2018). In this journal by (IEEE Staff, 2019), researchers explained that Round Robin is considered an efficient and fair algorithm because all processes are given the same amount of time.

Priority Scheduling Algorithm

The Priority Scheduling algorithm created by Kyle Singer et al. (2020) revealed that they wanted an interactive application that could provide fast response times because, usually, at the other end of the external interaction, there are users waiting for a response. Existing parallel platforms designed for multicore hardware do not work well for such interactive applications, as they are designed to maximize throughput, not responsiveness. Interactive applications may have a mix of interactive and computationally intensive tasks occurring simultaneously, so the scheduler must be able to differentiate and prioritize the tasks so that tasks that require a faster response are prioritized (Singer et al., 2020). In the journal (Jin & Yu, 2022a), researchers explained that an intelligent scheduling system is urgently needed to standardize operation management, intelligently schedule incoming and outgoing tasks, maximize the rational use of intelligent devices, and realize the automation of unmanned operations in actual scheduling situations.

Yet Another Fog Simulator (YAFS)

YAFS is a Fog Computing simulator tool designed to analyze application designs and incorporate strategies for placement, scheduling, and routing (Lera et al., 2019b). YAFS uses a generic library for the creation of discrete event simulation scenarios called Simpy. Simpy is a robust and reliable discrete-event simulator (DES) solution that includes tools for defining processes (active components) and shared resources (such as network links and queues). With this, it is possible to execute simulations in three modes: as fast as possible, in real time, or manually to step through events.

METHOD

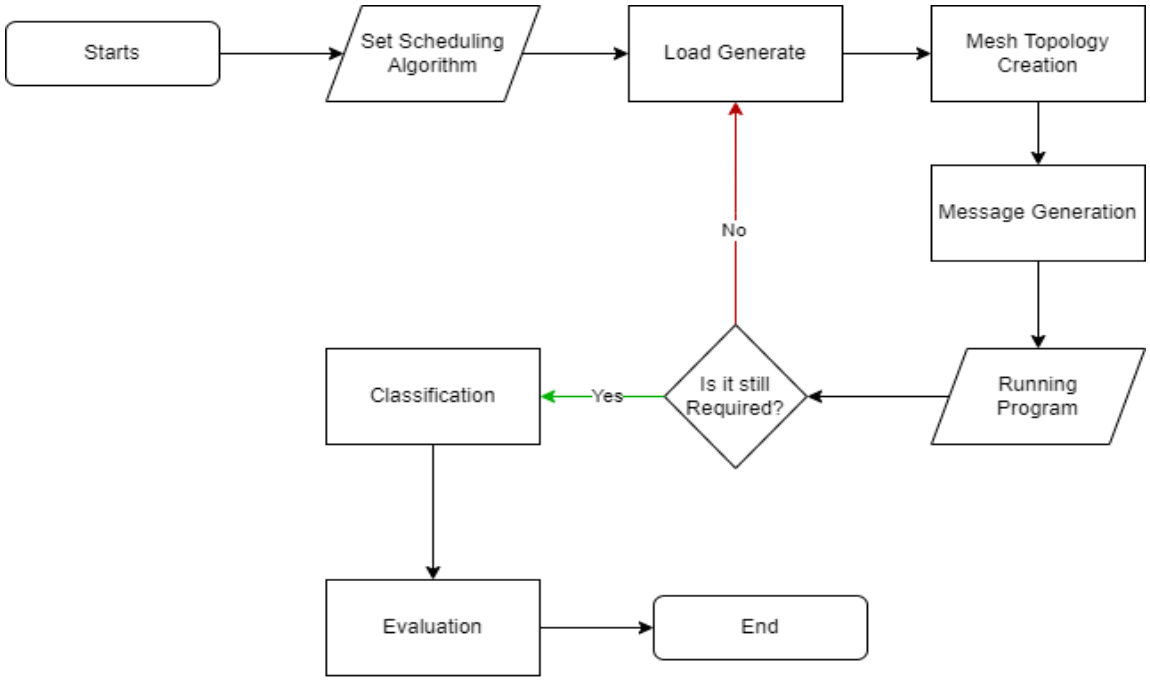

Fig. 1 YAFS Development System

The system built in this final project research is an implementation of the Scheduling algorithm on YAFS Tools. Figure 1 represents the development of the YAFS system.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i3.12682 p-ISSN : 2541-044X

Set Scheduling Algorithm

The algorithms used in this final project research consist of two, namely Round Robin and Priority Scheduling. To be maximally utilized, management and scheduling of Fog Computing resources such as storage, applications, and other services are needed (Mahmood Ibrahim et al., 2021). The scheduling algorithm is one of the algorithms used to organize a given load queue to be processed properly by Fog Computing. The dataset used is a short message sent through the YAFS simulator or tool. In the YAFS program, there are 3 modules: sensor, service (Fog), and Actuator. The three modules are used as modulators that are used as senders and receivers of the messages created. This research uses several modules as load enhancers or Load Balancing to test the performance of the algorithm.

Load Generator

Module addition is done to control or increase the load. This is intended to validate the scenario under test. To evaluate the scheduling algorithm's performance, the tested scenario needs a specified load.

Mesh Topology Creation

The creation of a mesh topology is meant to simplify the path the message takes to the destination module. A topology consists of a set of Nodes and Links. Nodes are network elements that function as a link between other elements and have the ability to run or host applications. In making a mesh topology, there are two variables, namely Entity and Link. The Entity variable indicates that the variable is used as a module, and the Link variable is a path or path that serves to be traversed by the message to get to the destination module.

Message Generator

The goal of message production is to provide the data needed to test the scenario. M.A. and M.B. are the two individuals who made up this message. Both communications have distinct objectives and directives. While message M.B is a message sent from Service A (Fog) to the actuator, message M.A is a message sent from the sensor to ServiceA (Fog). The generated Mesh Topology route is used to send the message. Following that, the message executes the supplied instruction in accordance with the number of loads that have been made.

Running Program

Running is executed by running the program if you have added modules and created the required topology and messages. The program generates a Graph and a CSV file from the results that have been made before. Both files contain the Mesh Topology and the time taken for the message to reach the destination module.

Classification

Classification is done to organize or group the results of Time Processing (time earned) during the process. The CSV file obtained by the program is converted into an Excel file. Because CSV files get less significant results. Therefore, the file must be converted and grouped into one file.

Evaluation

Evaluation is done by counting the Time Processing value from the time results obtained from the classification results. Time Processing is obtained by reducing the Time out of the M.B. message with the Time in of the M.A. message in a time unit of seconds (s). Then, Time Processing is calculated on average from the whole. Then, the results obtained from the average are the results of the previous program run.

RESULT

The primary test is to determine the scheduling algorithm's level of temporal complexity based on the number of loads. The simulation time is impacted by the addition of application modules to Fog. Since

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

more network connections are required for message exchange (Perez Abreu et al., 2020). Table 1 shows the performance of the algorithm in each scenario.

Table 1 Result Performance Algorithm

FOG NODE / DEVICES	4 FOG NODES		6 FOG NODES		8 FOG NODES	
	Round Robin	Priority Scheduling	Round Robin	Priority Scheduling	Round Robin	Priority Scheduling
4	35,336	33,322	39,168	33,432	41,149	35,795
6	79,562	85,799	98,831	105,926	90,418	103,921
8	89,667	87,136	71,495	104,157	132,434	125,785
Average (s)	68,188	68,752	69,831	81,171	88,000	88,500

The Round Robin and Priority Scheduling algorithms are used to conduct program testing with a preset load (number of devices). The purpose of this test is to evaluate a suitable system scheduling algorithm. The average processing time was discovered from the test using the designated Devices. When compared to Priority Scheduling, the Round Robin method produces a faster Processing Time.

This paper analyzes the test results of each scenario and found that when the number of Fog Nodes is greater than 2 and they are connected between Fog Nodes, the Sensors and Actuators can communicate through the existing Fog Nodes. The thing that affects Time Processing is the number of Devices. The more Devices used, the slower the processing. Meanwhile, placing multiple Fog Nodes results in faster processing time. When the simulator is run with different devices, the Processing Time obtained is the same, and there is no change in each scenario run. Because the simulator consistently replicates the same conditions and parameters in the simulation. The following is an analysis of the test results for each scenario.

Figure 2 is a test scenario in this research. This scenario is carried out to measure the performance of the scheduling algorithm by adding a predetermined load. The results of each scenario can prove to be a good scheduling algorithm to be tested in this case.

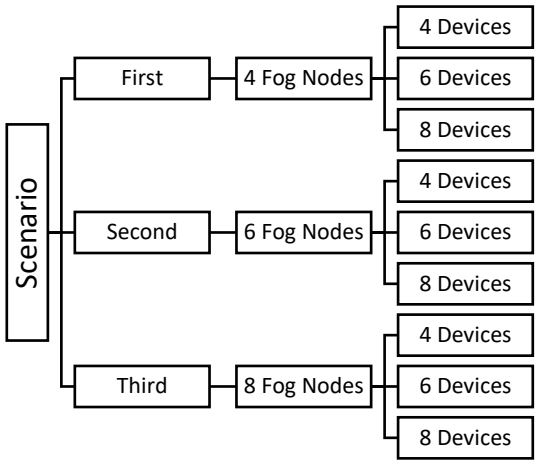

Fig. 2 Scenario Test

Fog Node is a distributed computing entity involved in collecting, processing, and analyzing data from connected Internet of Things (IoT) devices. Devices are connected devices, such as sensors and Actuators. The number of devices is divided equally, such as 4 devices (2 sensors and 2 actuators), 6 devices (3 sensors and 3 actuators), and 8 devices (4 sensors and 4 actuators).

The first scenario was done to compare processing times based on the amount of loads. Test is done by contrasting the impact of the quantity of Devices with the quantity of Fog Nodes. In this test, there were four fog nodes employed and four, six, and eight devices.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 2 Result Test Scenario 1

4 Fog Node		
Time Processing	Round Robin	Priority Scheduling
4 Devices	35,336	33,322
6 Devices	79,561	83,849
8 Devices	89,668	107,934

Table 2 shows the Time Processing results of the two algorithms tested on each Device. In the table, it can be seen that when testing with four Devices, the Priority Scheduling algorithm produces faster Time Processing than Round Robin. But when the number of Devices increases, the resulting processing time is higher. This is because the complexity level of Priority Scheduling is not as simple as the Round Robin algorithm.

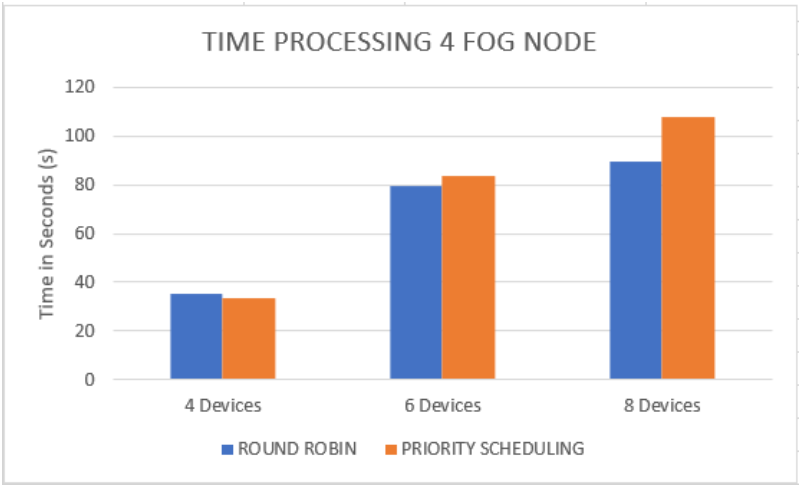

Fig. 3 Result 4 Fog Nodes

Figure 3 is a graph of the test results for scenario 1. The program runs Fog Nodes with 4 x 4 Devices, Fog Nodes with 4 x 6 Devices, and Fog Nodes with 4 x 8 Devices. Figure 2 shows the difference in Time Processing between the two scheduling algorithms. The Round Robin algorithm produces better Time Processing than Priority Scheduling. This is because Round Robin has a simple method and algorithm complexity, so it greatly affects Time Processing.

The graph and table 2 show that the Priority Scheduling algorithm has a high Time Processing value, averaging 75.036 seconds. With the average that Priority Scheduling has, the resources used cannot be properly maximized. Because some systems must be executed in the shortest possible time. The average processing time for Round Robin is 68.189 seconds, which is a low value. Results like this are necessary in computing because Round Robin's time result is shorter and the system is resource-efficient.

Table 3 Result Test Scenario 2

6 Fog Node		
Time Processing	Round Robin	Priority Scheduling
4 Devices	39,167	33,431
6 Devices	98,830	105,925
8 Devices	71,495	104,157

The second scenario, the Fog Nodes used increased to six Fog Nodes. Table 3 shows the results of each test in the previous scenario. The results of the test show that the Priority Scheduling algorithm is not suitable for the YAFS simulator or Tools. Because the mechanism of the Priority Scheduling algorithm requires a priority level in queuing, that's way, it takes a long time to process the queue

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

because the queue is processed based on the priority level given. In contrast to Round Robin, the mechanism of this algorithm is arguably quite simple because Round Robin's performance executes all data entered in the queue. Therefore, the time generated by Round Robin is shorter.

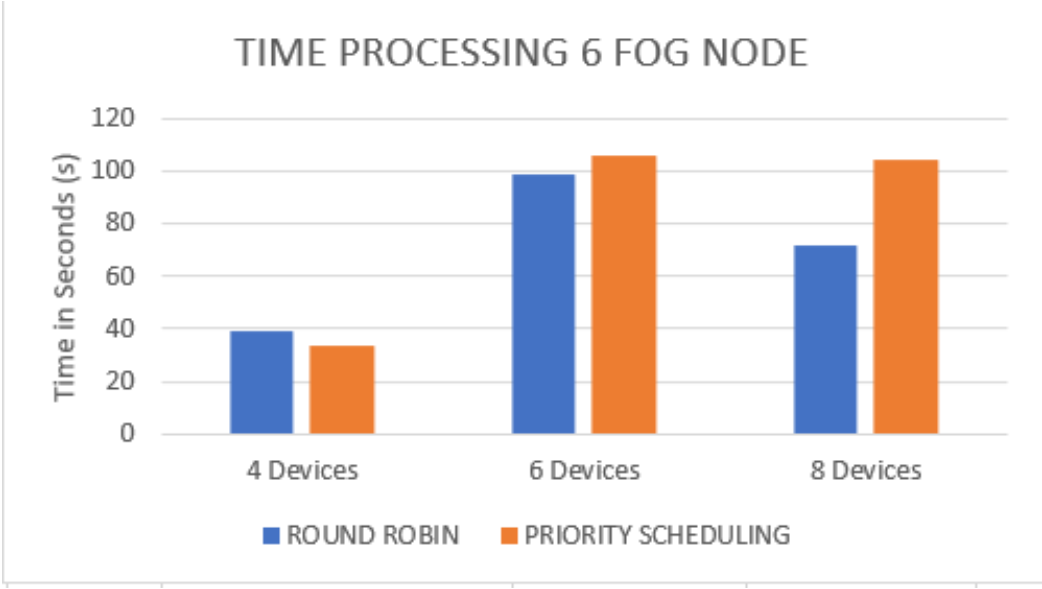

Fig. 4 Result 6 Fog Nodes

Figure 4 is a graph of the program results with six Fog Nodes. Fog nodes were tested with the first scenario. The Priority Scheduling algorithm still cannot get a better Processing Time than the round-robin algorithm. Because of the Priority algorithm, when the number of Devices is added, the resulting Processing Time becomes longer than Round Robin. Priority Scheduling gets an average Time Processing result of 81.171 seconds. The resulting Time Processing is higher than the previous scenario; this cannot be called the best scheduling algorithm because the time obtained is irrelevant. Round Robin produces Time Processing with an average of 69.831 seconds. With these results, the system can process requests faster due to Round Robin's favorable performance in managing the given load.

Table 4 Result Test Scenario 3

8 Fog Node		
Time Processing	Round Robin	Priority Scheduling
4 Devices	41,148	35,794
6 Devices	90,418	103,921
8 Devices	132,434	125,785

The last scenario test, table 4 shows the test to get Time Processing results with 8 Fog Nodes and tested with the same Devices. This test is conducted to determine the number of Fog Nodes (if any) and the number of Devices that have been determined. The better the processing time results, or otherwise.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

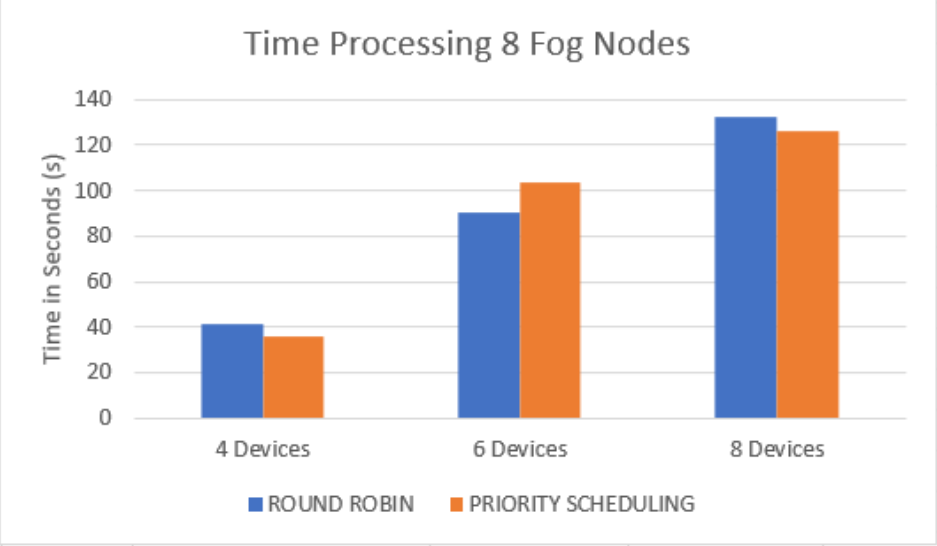

Fig. 5 Result 8 Fog Node

Figure 5 shows the Processing Time generated from scenario 3. The performance of the scheduling algorithm for 8 Fog Nodes with various devices in the graph states that the Round Robin algorithm has a significantly faster or more efficient time than the Priority Scheduling algorithm. Because Round Robin produces a time that is very relevant to the number of Fog Nodes and Devices used as a load, The Round Robin algorithm obtains a very favorable time to be used on the system.

Priority Scheduling shows significant Time Processing results because, when tested, the Time Processing results increase according to the number of Fog Nodes and Devices. The average obtained from the test results is 88.500 seconds. Priority Scheduling performance still cannot be used because it cannot maximize resources on the system because the resulting time is higher and can slow down the system process.

Round Robin in the case of scenario 3 shows that the Fog Nodes work well in sending the message. The 8 interconnected Fog Nodes and Devices that have been determined get significant Time Processing results. The average obtained is 88,000 seconds. When the simulator is run by the researcher's device, the results obtained are as maximum as possible because the researcher runs it more than once. So, it can be seen from testing the two scheduling algorithms, still in favor of the Round Robin algorithm as a good scheduling algorithm for the system.

DISCUSSIONS

The scheduling algorithm's impact on the load generator

The scheduling algorithm's effectiveness is greatly impacted by the amount of load applied to evaluate it. The processing time that results from an increase in Devices may have an impact on the scheduling algorithm's complexity. The two scheduling algorithms have different mechanisms and different kinds of performance. In the test case, Round Robin has a simpler performance to be examined. Compared to priority scheduling, round robin has lower processing time and higher time efficiency. This is due to a process in Priority Scheduling that must be used based on Priority. The processed data can then move more slowly as a result of the processing. So, Round Robin is an appropriate scheduling algorithm for this case.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 3, July 2023
DOI : https://doi.org/10.33395/sinkron.v8i3.12682
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Evaluate Round Robin over Priority Scheduling

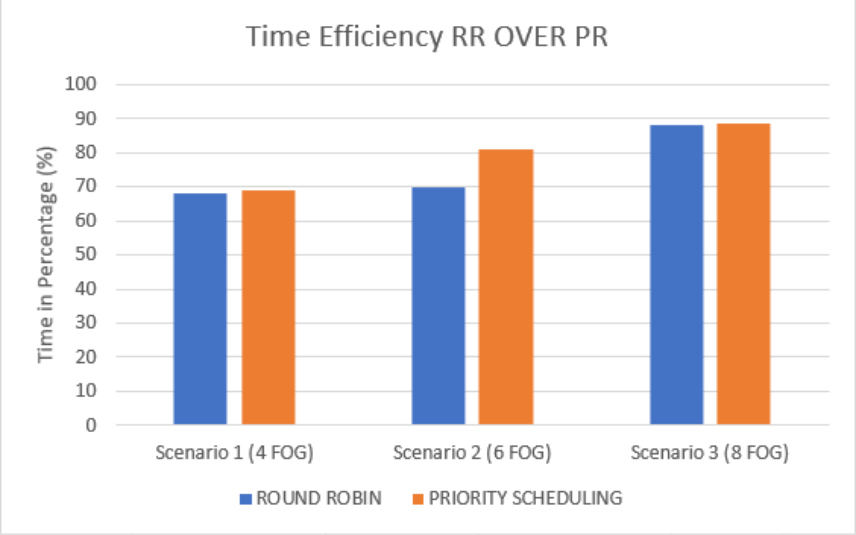

Fig. 6 Time Efficiency RR Over PR

Figure 6 shows the results of the three scenarios tested to get the Time Efficiency of the Round Robin algorithm with Priority Scheduling. Time Efficiency is the amount of time generated by the two algorithms to measure the level of complexity of the scheduling algorithm. The highest Round Robin Time Efficiency against Priority Scheduling is 11%. The Round Robin algorithm is more efficient for this simulator. The resulting small difference can affect the processing to get a lower Processing Time. Because some IoT devices require Real-time. So, the scheduling needed for this simulator is the Round Robin algorithm.

CONCLUSION

Based on the test results to get good performance from the Round Robin and Priority Scheduling scheduling algorithms with YAFS tools, it can be concluded that the Round Robin algorithm is a simple scheduling algorithm that can produce Time Processing that can be used in load balancing. The scheduling algorithm can be said to be the best if the resulting processing time is getting smaller. With small processing results, the system can maximize existing resources. In addition to Time Processing, there is Time Efficiency between the two algorithms to measure the level of complexity of the two algorithms. Testing the first, second, and third scenarios has shown that the Round Robin algorithm has an advantage over Priority Scheduling in terms of better performance. Time Processing produced by Round Robin produces more efficient results, and the highest Time Efficiency that Round Robin has against Priority Scheduling is 11%. Due to the complexity of the Round Robin algorithm, it is simpler to execute all the data in the queue, and some IoT devices require Real-Time processing. The number of Devices used can affect the level of processing. The three scenarios can be interpreted so that Load Generate can affect the resulting Time Processing. The scheduling algorithm is tested with the same Devices or Nodes and produces different Time Processing for each scenario. Each increase in the number of Fog Nodes and Devices that are then tested by the scheduling algorithm has a very large impact on achieving a very short Processing Time. So, the Round Robin algorithm falls into the best scheduling category for this case.

REFERENCES

Abdali, T. A. N., Hassan, R., Aman, A. H. M., & Nguyen, Q. N. (2021). Fog Computing Advancement: Concept, Architecture, Applications, Advantages, and Open Issues. *IEEE Access*, *9*, 75961–75980. https://doi.org/10.1109/ACCESS.2021.3081770

Choudhari, T., Moh, M., & Moh, T. S. (2018). Prioritized task scheduling in fog computing. *Proceedings of the ACMSE 2018 Conference*, *2018-January*. https://doi.org/10.1145/3190645.3190699

*name of corresponding author

de Donno, M., Tange, K., & Dragoni, N. (2019). Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog. *IEEE Access*, *7*, 150936–150948. https://doi.org/10.1109/ACCESS.2019.2947652

Guevara, J. C., & da Fonseca, N. L. S. (2021). Task scheduling in cloud-fog computing systems. *Peer-to-Peer Networking and Applications*, *14*(2), 962–977. https://doi.org/10.1007/s12083-020-01051-9

IEEE Staff. (2019). *2019 SoutheastCon*. IEEE.

Jin, X., & Yu, L. (2022a). Research and implementation of high priority scheduling algorithm based on intelligent storage of power materials. *Energy Reports*, *8*, 398–405. https://doi.org/10.1016/j.egyr.2022.03.126

Jin, X., & Yu, L. S. (2022b). Research and implementation of high priority scheduling algorithm based on intelligent storage of power materials. *Energy Reports*, *8*, 398–405. https://doi.org/10.1016/j.egyr.2022.03.126

Lera, I., Guerrero, C., & Juiz, C. (2019a). YAFS: A Simulator for IoT Scenarios in Fog Computing. *IEEE Access*, *7*, 91745–91758. https://doi.org/10.1109/ACCESS.2019.2927895

Lera, I., Guerrero, C., & Juiz, C. (2019b). YAFS: A Simulator for IoT Scenarios in Fog Computing. *IEEE Access*, *7*, 91745–91758. https://doi.org/10.1109/ACCESS.2019.2927895

Mahmood Ibrahim, I., MSadeeq, M. A., M Zeebaree, S. R., Shukur, H. M., Jacksi, K., Radie, A. H., Maseeh Yasin, H., & Najat Rashid, Z. (2021). Task Scheduling Algorithms in Cloud Computing: A Review. In *Turkish Journal of Computer and Mathematics Education* (Vol. 12, Issue 4).

Margariti, S. v., Dimakopoulos, V. v., & Tsoumanis, G. (2020). Modeling and simulation tools for fog computing-A comprehensive survey from a cost perspective. *Future Internet*, *12*(5). https://doi.org/10.3390/FI12050089

Murad, S. A., Muzahid, A. J. M., Azmi, Z. R. M., Hoque, M. I., & Kowsher, M. (2022). A review on job scheduling technique in cloud computing and priority rule based intelligent framework. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 34, Issue 6, pp. 2309–2331). King Saud bin Abdulaziz University. https://doi.org/10.1016/j.jksuci.2022.03.027

Neware, R. (2019a). *Fog Computing Architecture, Applications and Security Issues: A Survey*. https://doi.org/10.20944/preprints201903.0145.v1

Neware, R. (2019b). *Fog Computing Architecture, Applications and Security Issues: A Survey*. https://doi.org/10.20944/preprints201903.0145.v1

Perez Abreu, D., Velasquez, K., Curado, M., & Monteiro, E. (2020). A comparative analysis of simulators for the Cloud to Fog continuum. *Simulation Modelling Practice and Theory*, *101*. https://doi.org/10.1016/j.simpat.2019.102029

PHORNCHAROEN, S., & SA-NGIAMVIBOOL, W. (2018). A proposed round robin scheduling algorithm for enhancing performance of CPU utilization. *Przeglad Elektrotechniczny*, *94*(4), 34–38. https://doi.org/10.15199/48.2018.04.07

Sabireen, H., & Neelanarayanan, V. (2021). A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges. *ICT Express*, *7*(2), 162–176. https://doi.org/10.1016/j.icte.2021.05.004

Sakshi, Sharma, C., Sharma, S., Kautish, S., A. M. Alsallami, S., Khalil, E. M., & Wagdy Mohamed, A. (2022). A new median-average round Robin scheduling algorithm: An optimal approach for reducing turnaround and waiting time. *Alexandria Engineering Journal*, *61*(12), 10527–10538. https://doi.org/10.1016/j.aej.2022.04.006

Singer, K., Goldstein, N., Muller, S. K., Agrawal, K., Lee, I. T. A., & Acar, U. A. (2020). Priority Scheduling for Interactive Applications. *Annual ACM Symposium on Parallelism in Algorithms and Architectures*, 465–477. https://doi.org/10.1145/3350755.3400236

Singh, J., Singh, P., & Gill, S. S. (2021). Fog computing: A taxonomy, systematic review, current trends and research challenges. In *Journal of Parallel and Distributed Computing* (Vol. 157, pp. 56–85). Academic Press Inc. https://doi.org/10.1016/j.jpdc.2021.06.005

Tychalas, D., & Karatza, H. (2020). An Advanced Weighted Round Robin Scheduling Algorithm. *ACM International Conference Proceeding Series*, 188–191. https://doi.org/10.1145/3437120.3437304

*name of corresponding author