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).

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).