Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Dynamic Task Scheduling for Optimal Resource Utilization in Cloud Computing Environments using Johnson Sequencing Algorithm

Pallab Banerjee[1], Sharmistha Roy [2], T.Hemalatha[3], Niva Tripathy[4], Anurag Sinha[*], Mohan Kumar Dehury[5], Pranav Joshi[6]

[1,2] Usha Martin University, Ranchi, Jharkhand, India.
[3]Assistant Professor, Dept of CSE, Koneru Lakshmaiah Education Foundation, Green fields, Vaddeswaram, Guntur-522302
[4]Driems Autonomous Engineering College, Cuttack, Tangi nivatripathy@driems.ac.in
[*]School of Computing and Information Science, IGNOU, New Delhi, India
[5]Amity University Jharkhand, Ranchi, India.
[6]Graphic era (deemed to be university)

*Corresponding author : anuragsinha257@gmail.com

***Abstract*—CPU scheduling is among the most important operations operated by an operating system (OS). There are multiple CPU scheduling methods that are present, one of them are Round Robin(RR) which is an optimized in time shared atmosphere. The efficiency of Round Robin (RR) is totally dependent upon the selection of Time Quantum (*TQ*). In Current paper an more optimized CPU scheduling algorithm MARR (Mid Average Round Robin) will been proposed of Round Robin that will be implementing on Johnson Sequencing Algorithm. that use the dynamic time quantum in place of the static time quantum used in Round Robin. The performance of this proposed algorithm is experimentally compared with FCFS, normal Johnsons sequencing, and MARR on Johnsons sequencing. The outcomes of the approach described in the current paper show optimized performance in terms of reduced service time, which was demonstrated using graphs and tables.**

***Index Terms*—Internet of Things, Machine Learning; Cyber security;Anomaly detection.**

I. INTRODUCTION

A game-changing technology that provides scalable, on-demand access to computer resources through the Internet is cloud computing. It has completely changed how organisations and people use and manage their computing workloads. Effective task scheduling, which includes allocating computer resources to activities or workloads to maximise performance and resource utilisation, is one of the major issues in cloud computing [1]. Task scheduling is essential for optimising the usage of cloud resources, attaining high system performance, and maintaining user fairness. It entails making informed judgements about where and when to carry out activities, taking into account variables including workload characteristics, task dependencies, and resource availability [2].

Task scheduling is a challenging challenge in cloud computing systems because of a number of distinctive features. First off, the cloud is made up of a variety of resources, such as virtual machines, containers, and storage, all of which have unique properties. The complexity of job distribution may increase if these resources are dispersed across many data centres or geographical regions. Second, the dynamic nature of cloud workloads makes task scheduling much more difficult [3]. As workloads change over time due to demand peaks and valleys, the scheduler must adjust and allot resources appropriately. Tasks may also be interdependent, necessitating careful planning and scheduling to guarantee proper execution. The requirement to uphold service-level agreements (SLAs) and maximise performance goals is another crucial feature of job scheduling in cloud computing. SLAs outline the quality of service standards that the cloud provider is required to uphold, including response time, throughput, and availability. Algorithms for scheduling must take these goals into account while effectively using available resources to save costs and increase customer satisfaction. To overcome these difficulties, effective task scheduling algorithms and techniques have been created. Heuristics and algorithms based on queuing theory, which are traditional scheduling approaches, are frequently employed [4]. However, there is a requirement for more intelligent and adaptable scheduling systems due to the complexity and size of cloud settings.

Recent developments in optimisation and machine learning methods have created new possibilities for work scheduling in cloud computing. By learning from previous data and adjusting to shifting workload conditions, techniques including reinforcement learning, deep learning, and evolutionary computing have showed promise in improving scheduling decisions. In the end, scheduling tasks is an essential component of cost effectiveness, performance optimisation, and resource management in cloud computing [5]. It entails deliberative resource allocation decisions that take into account a variety of aspects, including workload characteristics, resource availability, and performance goals. High system performance, SLA compliance, and optimal resource utilisation all depend on effective scheduling methods and techniques [6]. Future developments in cloud computing will be fueled by research and innovation in this field, making it possible for task scheduling systems to be more scalable, adaptable, and effective.

Nomenclatures:

M	Number of virtual machine (servers)
E(S)	Average service time
K	Maximum capacity of the system
L_q	Avg. no. of tasks in the queue
L_s	Avg. no. of tasks in the system
P(x)	Probability of x arrivals ($x = 0, 1, 2, \ldots$)
W_q	Avg. waiting time in the queue
W_s	Avg. waiting time in the system
X	Number of arrivals per unit of time
λ	Avg. Arrival rate
μ	Avg. service rate

TABLE I
TERMS RELATED TO SCHEDULING

II. REVIEW OF LITERATURE

"A Survey of Task Scheduling in Cloud Computing: This thorough review offers taxonomy of cloud computing work scheduling techniques. It divides the algorithms into several categories depending on different factors including task models, scheduling goals, and optimisation methods. The survey examines the benefits and drawbacks of current algorithms and proposes new areas for further study to improve work scheduling in cloud systems [7].

"Energy-Efficient Task Scheduling in Cloud Computing: A Review" by Gupta et al. in [8]: The article addresses cloud computing's energy-efficient work scheduling techniques. It provides a summary of energy-aware scheduling approaches and how they affect cloud data centres' power usage. The paper outlines the difficulties and possibilities of creating energy-efficient scheduling algorithms and offers suggestions for potential remedies.

"Deadline-Constrained Task Scheduling in Cloud Computing: A Comprehensive Review" by Deng et al. in [9]: The work scheduling issue with deadline restrictions in cloud computing is addressed in this overview study. It provides an overview of the literature on deadline-aware scheduling algorithms and covers the approaches for dealing with jobs that have a deadline. The study offers a variety of scheduling methods and tactics for keeping to deadlines while making the most use of available resources.

"Task Scheduling Algorithms in Cloud Computing: A Systematic Review" by Kumar et al. in [10]: The most recent cloud computing job scheduling techniques are examined in this systematic study. It divides the algorithms into categories according to the goals of their optimisation, such as makespan reduction, load balancing, and energy efficiency. The study uses parameters like execution time, resource utilisation, and scalability to assess the effectiveness of various algorithms.

"Machine Learning-Based Task Scheduling in Cloud Computing: A Survey" by Wang et al. in [11]: This study addresses methods for cloud computing work scheduling that leverage machine learning. It gives an overview of several machine learning approaches used for task scheduling, such as reinforcement learning, genetic algorithms, and neural networks. The poll talks about the benefits, difficulties, and potential future uses of machine learning in scheduling.

"Multi-Objective Task Scheduling in Cloud Computing: A Review" by Chen et al. in [12]: The present paper investigates multi-objective task scheduling in cloud computing, taking into account many competing goals such as making the most efficient use of resources while minimising makespan and energy usage. It provides a thorough examination of the optimisation techniques, assessment measures, and multi-objective scheduling algorithms currently in use. The assessment points up areas for more investigation as well as prospective directions.

"Task Scheduling Techniques for Big Data Analytics in Cloud Computing: A Review" by Shahzad et al. in [13]: This evaluation focuses on job scheduling strategies created especially for cloud computing's big data analytics. The difficulties and needs of scheduling data-intensive jobs, including data localization, load balancing, and fault tolerance, are covered. The paper gives a summary of the current scheduling techniques and emphasises the significance of taking big data workload characteristics into account.

These reviews of the literature offer insightful analyses of the task scheduling issue in cloud computing, covering a range of topics including multi-objective optimisation, big data analytics, energy efficiency, deadline restrictions, and optimisation objectives. These evaluations may be used by researchers to gain insight into the status of the field, pinpoint areas for further study, and create new job scheduling algorithms and approaches for effective resource management in cloud settings.

III. SCHEDULING ALGORITHM

Round Robin (RR) The conventional, basic, and generally utilized time-shared planning calculation is called Round Robin (RR). It is almost indistinguishable from the First Come First Served ((FCFS) planning strategy, besides change between processes, a right of first refusal is added [14]. The most limited time cut utilized for Round Robin (RR) sequencing techniques is alluded to as the Time Quantum (TQ). Bypassing the Ready Queue (RQ), the computer chip scheduler appropriates the computer processor to each interaction using his dispatcher at timespans to one Time Quantum (TQ) [15]. Whenever one more cycle shows up, it is added to the furthest limit of the round Line. The computer processor scheduler picks the main errand from the prepared line, and sets the clock that hinders after a Period Quantum(TQ), and conveys the cycles [16].

The computer chip suspends the cycle and adds it to the furthest limit of the round line when the TQ lapses. A cycle joyously assumes command over the computer processor when it goes past the constraint of its transient quantum (TQ). By altering the time quantum (TQ) according to the assortment of dynamic cycles in the prepared line, I endeavored to address the time quantum issue progressively in this paper [16], [17].

IV. OBJECTIVE

The system in the recent study was developed. Using two cloud servers, the Dynamic Johnson Sequencing method has

been used to reduce service time in cloud settings. A Gantt chart makes it extremely straightforward to determine service time when taking into account a bundle of several works [8]. The service times for each job may be obtained utilising that Gantt chart in order to do that [9]. The system's common range of customers in the line and inside the machine, as well as the average waiting time inside the machine and in the line, could be decreased with the implementation of Dynamic Johnson's sequencing rule [18].

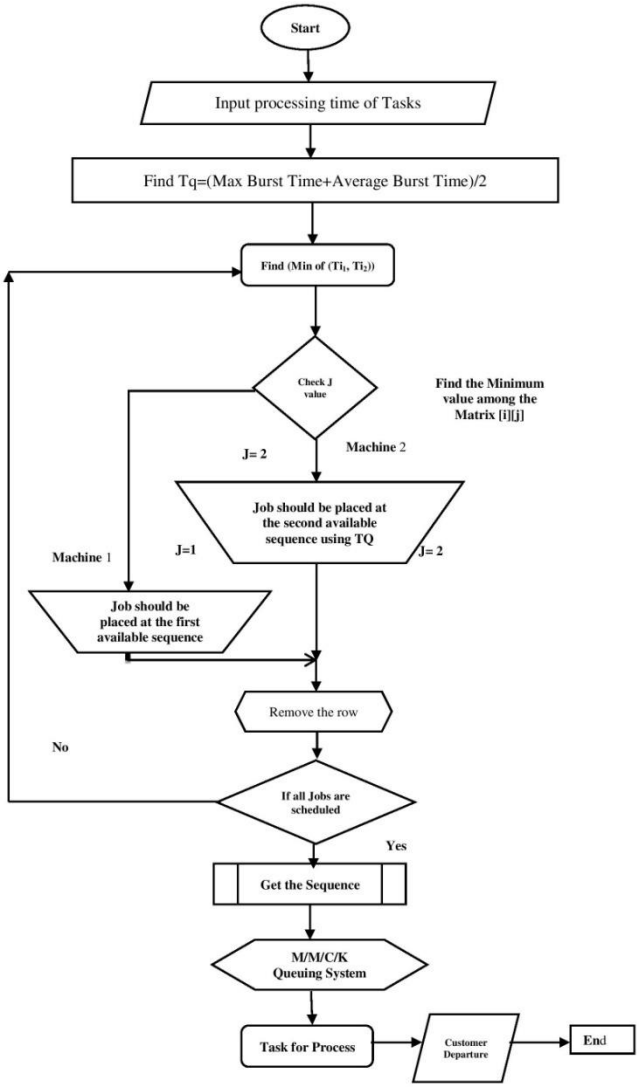

Fig. 1. Flowchart of proposed algorithm

V. PROPOSED ALGORITHM

In this paper we will be minimizing the output of our system for better performance by using different algorithms and after that we will compare the Dynamic Johnson Sequencing algorithm with algorithm like FCFS and Johnsons Sequencing as shown in Table 2, 3, 4,and 6.

TABLE II
NUMERICAL ANALYSIS FOR FCFS ALGORITHM

JOBS	Server 1	Server 2
A	0.06	0.02
B	0.08	0.08
C	0.03	0.01
D	0.04	0.05
E	0.07	0.03

Fig. 2. Gantt Chart of FCFS

A. Implementation with FCFS Scheduling using Two server

Presently, Administration time for each cycle can be find out by utilizing the above Gantt graphs of FCFS (Table 6), and the Assistance time and correspondingly Normal help rate can be determined.

FOR JOBS
A= (0.08-0) = 0.08
B= (0.22-0.06) = 0.16
C= (0.23-0.14) = 0.09
D= (0.28-0.17) = 0.11
E= (0.31-0.21) = 0.10
T = 0.54
Service time $=\frac{0.54}{5}$ = 0.108.
Service rate $(\mu)=\frac{1}{E(s)}$ = 9.259259

Presently, we will consider the Johnson Sequencing calculation, Administration rate can be determined as displayed on Table 7.
D= (0.09-0) = 0.09
B= (0.20-0.04) = 0.16
E= (0.23-0.12) = 0.11
A= (0.27-0.19) = 0.08
C= (0.29-0.25) = 0.04
T=0.48
Service time = $\frac{0.48}{5}$ = 0.096.
Service rate $(\mu)=\frac{1}{E(s)}$ = 10.416666

Again we will consider the MARR Calculation in Johnson Sequencing calculation, Avg administration rate can be determined as displayed on (Table 9).
D= (0.04-0) = 0.04
B= (0.12-0.04) = 0.08
E= (0.21-0.12) = 0.09
A= (0.31-0.21) = 0.10
C= (0.47-0.31) = 0.16
T=0.47
Service time = $\frac{0.47}{5}$ = 0.094.
Service rate$(\mu)=\frac{1}{E(s)}$ = 10.638297

B. Implementation with Johnson Sequencing

BT=M1+M2

TABLE III
PROCESSING TIME OF DIFFERENT JOBS

Jobs	Server 1	Server 2	Burst Time(BT)
A	0.06	0.02	0.08
B	0.08	0.08	0.16
C	0.03	0.01	0.04
D	0.04	0.05	0.09
E	0.07	0.03	0.1

TABLE IV
IN AND OUT TIME OF TWO SERVERS IN JOHNSON SEQUENCING

Jobs	Server M1		Server M2	
	In-Time	Out-Time	In-Time	Out-Time
D	0	0.04	0.04	0.09
B	0.04	0.12	0.12	0.2
E	0.12	0.19	0.2	0.23
A	0.19	0.25	0.25	0.27
C	0.25	0.28	0.28	0.29

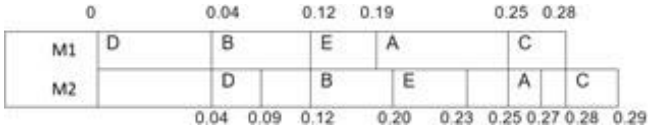

Fig. 3. Gantt chart with Johnsons Sequencing

C. Implementation Of Modified Johnson Sequencing Using Dynamic Round Robin Technique

For taking Time Quantum first of all we will sort the process. Then We will Find the Mid point, after finding mid, we will count the further process burst time (BT) and add them, then we will divide those process to get Time Quantum with the help of MARR and implement that TQ on Johnsons sequencing Algorithm [19].

To find first Mid we will do certain mathematical operation:

MID= First process index + Last process index/2

Therefore,

MID= $\frac{0+4}{2}$

Therefore, MID= 2

C = 4

A = 8

D = 0.09 ← MID

E = 0.10

B = 0.16

TQ= $\frac{0.9+0.10+0.16}{3}$

TQ=0.12

While comparing the Gantt chart of FCFS, JS and MDJS in Fig.1, 2 and 3 shows that MDJS gives better performance. Following are the equations used in this paper:

TABLE V
IN AND OUT TIME OF TWO SERVERS IN MODIFIED JOHNSON SEQUENCING

Jobs	Server M1		Server M2	
	In-Time	Out-Time	In-Time	Out-Time
C	0	0.04	Ideal	Ideal
A	0.04	0.12	Ideal	Ideal
D	0.12	0.21	Ideal	Ideal
E	0.21	0.31	Ideal	Ideal
B	0.31	0.43	0.43	0.47

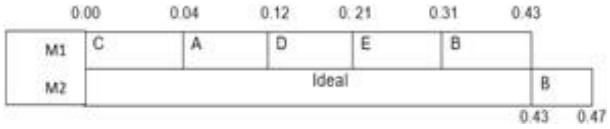

Fig. 4. Gantt Chart using MDJS Algorithm in Johnsons Sequencing.

Average Service time:

$$E(S) = \frac{\sum_{i=0}^{n} S_i}{n} \quad (1)$$

Service rate will be:

$$\mu = \frac{1}{E(S)} \quad (2)$$

Probability that the system is idle :

$$P_0 = \left[\sum_{n=0}^{S-1} \frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n + \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S \frac{S\mu}{S\mu-\lambda}\right]^{-1} \quad (3)$$

Average number of tasks waiting in the queue:

$$L_q = \left[\frac{1}{S-1!} \times \left(\frac{\lambda}{\mu}\right)^S \times \frac{\lambda\mu}{(S\mu-\lambda)^2}\right] \times P_0 \quad (4)$$

Average number of tasks in the system:

$$L_s = L_q + \frac{\lambda}{\mu} \quad (5)$$

Average waiting time of tasks in queue

$$W_q = \frac{L_q}{\lambda} \quad (6)$$

Average waiting time of tasks in the system

$$W_s = W_q + \frac{1}{\mu} \quad (7)$$

The outcomes are displayed in Tables VI, VII and VIII the particular recipe have been displayed in this ongoing paper. The outcomes will show the help time and avg holding up time can be advanced by executing MARR involving TQ in Johnson's Sequencing Calculation and holding framework in contrast with existing Johnsons' sequencing Calculation and FCFS Calculation [20]. As indicated by the mathematically results we have shown the correlation of L_q, L_s, W_q, and W_s. Table 7, 8, 9 show that the typical number and the typical

holding up time in the line and in the framework can be limited involving MDJS in Johnson sequencing calculation than FCFS and Johnsons Sequencing algorithm [21]. These examination gives improved results in the event of avg number of client and the typical holding up time if there should arise an occurrence of Johnson Sequencing Algorithm [22]. This study assists the CSP's with giving better nature of administration as holding up time is less and prompts consumer loyalty [23].

VI. COMPARATIVE ANALYSIS AND RESULTS

TABLE VI
RESULTS BY USING FCFS

	Lq	**Ls**	**Wq**	**Ws**
$\lambda = 2$	0.002496	0.218496	0.001248	0.109248
$\lambda = 4$	0.025854	0.457854	0.006463	0.114463
$\lambda = 6$	0.291191	1.047191	0.041598	0.149598

TABLE VII
RESULTS BY USING JOHNSONS SEQUENCING ALGORITHM

	Lq	**Ls**	**Wq**	**Ws**
$\lambda = 2$	0.001785	0.193785	0.000892	0.096892
$\lambda = 4$	0.017884	0.401884	0.004471	0.100471
$\lambda = 6$	0.141928	0.813928	0.020275	0.116275

TABLE VIII
RESULTS BY USING MDJS SCHEDULING ALGORITHM IN JOHNSONS SEQUENCING ALGORITHM

	Lq	**Ls**	**Wq**	**Ws**
$\lambda = 2$	0.001675	0.218496	0.001248	0.109248
$\lambda = 4$	0.016691	0.392691	0.004172	0.098172
$\lambda = 6$	0.131003	0.789003	0.018714	0.112714

VII. COMPARATIVE ANALYSIS WITH GRAPHS

Fig. 5. Average number of jobs in the queue (L_q).

From Fig. 5, 6, 7 and 8 it shows that the parameters L_q, L_s, W_q and W_s are less in MDJS as compared to FCFS and JS. So MDJS perform faster than FCFS and JS.

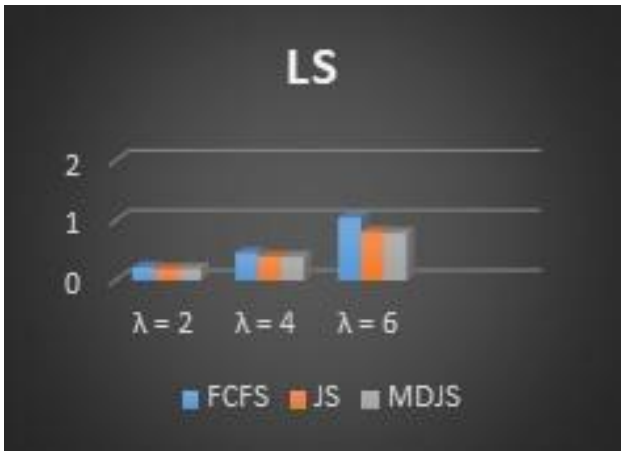

Fig. 6. Average number of jobs in the system (L_S).

Fig. 7. Average number of jobs in the queue (W_q).

Fig. 8. Average number of jobs in the system (W_S).

VIII. STATISTICAL ANALYSIS USING T-TEST

The results of a paired t-test, which was conducted to determine whether there are any significant differences between Mean, Std. Deviation, Std. Means of Error, Correlation, p-value and t-test value obtained from the proposed algorithm DMJS and those from other algorithms, including SJF, Johnson Sequencing using two servers and DMJS using two servers, using similar stopping standards for all task instances, are presented in Tables IX to XX. The acceptable p-value ought to be lower than alpha ($\alpha < 0.5$). The p-values for nearly all of the datasets, as shown in the tables below, were smaller than this alpha value, demonstrating a significant improvement in the proposed DMJS method over the other algorithms used for comparison. It follows that the DMJS performs better than alternative job scheduling algorithms in terms of Mean, Std. Deviation, Std. Means of Error, Correlation, p-value and t-test value.

TABLE IX
T-TEST RESULTS FOR L_q IN TERMS OF MEAN, STD. DEVIATION AND STD. MEANS OF ERROR

L_q (Average number of jobs in the queue)		Mean	Std. Deviation	Std. Error Mean
Pair 1	FCFS	0.10651367	0.160361115	0.092584533
	MDJS(2 Server)	0.04978967	0.070732413	0.040837377
Pair 1	JS (Server)	0.05386567	0.076687844	0.044275747
	MDJS(2 Server)	0.04978967	0.070732413	0.040837377

TABLE X
T-TEST RESULTS FOR L_q IN TERMS OF CORRELATION AND P-VALUE

L_q (Average number of jobs in the queue)		Correlation	One-Sided p Value
Pair 1	FCFS & MDJS(2 Server)	0.999	0.021
Pair 1	JS(2 Server) & MDJS(2 Server)	1	0.001

TABLE XI
T-TEST RESULTS FOR L_S IN TERMS OF MEAN, STD. DEVIATION AND STD. MEANS OF ERROR

L_S (Average number of jobs in the queue)		Mean	Std. Deviation	Std. Error Mean
Pair 1	FCFS	0.57451367	0.426486763	0.246232247
	MDJS(2 Server)	0.46673	0.29237115	0.168800562
Pair 1	JS(2 Server)	0.46986567	0.315611257	0.182218244
	MDJS(2 Server)	0.46673	0.29237115	0.168800562

TABLE XII
T-TEST RESULTS FOR L_S IN TERMS OF CORRELATION AND P-VALUE

L_S (Average number of jobs in the queue)		Correlation	One-Sided p Value
Pair 1	FCFS & MDJS(2 Server)	1.000	0.011
Pair 1	JS(2 Server) & MDJS(2 Server)	0.999	0.021

TABLE XIII
T-TEST RESULTS FOR W_q IN TERMS OF MEAN, STD. DEVIATION AND STD. MEANS OF ERROR

W_q (Average number of jobs in the queue)		Mean	Std. Deviation	Std. Error Mean
Pair 1	MDJS(2 Server)	0.00804467	0.009354862	0.005401032
	FCFS	0.01643633	0.021946097	0.012670585
Pair 1	MDJS(2 Server)	0.00804467	0.009354862	0.005401032
	JS(2 Server)	0.00854600	0.010314039	0.005954813

TABLE XIV
T-TEST RESULTS FOR W_q IN TERMS OF CORRELATION AND P-VALUE

W_q (Average number of jobs in the queue)		Correlation	One-Sided p Value
Pair 1	FCFS & MDJS(2 Server)	0.999	0.024
Pair 1	JS(2 Server) & MDJS(2 Server)	1.000	0.011

TABLE XV
T-TEST RESULTS FOR W_S IN TERMS OF MEAN, STD. DEVIATION AND STD. MEANS OF ERROR

W_S (Average number of jobs in the queue)		Mean	Std. Deviation	Std. Error Mean
Pair 1	FCFS	0.12443633	0.021946097	0.012670585
	MDJS(2 Server)	0.10671133	0.007595620	0.004385333
Pair 1	JS(2 Server)	0.10454600	0.010314039	0.005954813
	MDJS(2 Server)	0.10671133	0.007595620	0.004385333

TABLE XVI
T-TEST RESULTS FOR W_S IN TERMS OF CORRELATION AND P-VALUE

W_S (Average number of jobs in the queue)		Correlation	One-Sided p Value
Pair 1	FCFS & MDJS(2 Server)	0.593	0.596
Pair 1	JS(2 Server) & MDJS(2 Server)	0.548	0.631

TABLE XVII
T-TEST RESULTS FOR L_q IN TERMS OF PAIRED DIFFERENCES

L_q (Average number of jobs in the queue)		Paired Differences					t-Test Value
		Mean	Std. Deviation	Std. Error Mean	50% Confidence Interval of the Difference		
					Lower	Upper	
Pair 1	FCFS - MDJS(2 Server)	0.056724	0.08969948	0.051788019	0.01443926	0.09900874	1.095
Pair 1	JS(2 Server) - MDJS(2 Server)	0.004076	0.00595607	0.003438741	0.00126828	0.00688372	1.185

TABLE XVIII
T-TEST RESULTS FOR L_s IN TERMS OF PAIRED DIFFERENCES

L_s (Average number of jobs in the queue)		Paired Differences					t-Test Value
		Mean	Std. Deviation	Std. Error Mean	50% Confidence Interval of the Difference		
					Lower	Upper	
Pair 1	FCFS - MDJS(2 Server)	0.107783667	0.134267091	0.077519141	0.044489553	0.171077780	1.390
Pair 1	JS(2 Server) - MDJS(2 Server)	0.003135667	0.025366348	0.014645268	-0.008822144	0.015093478	0.214

TABLE XIX
T-TEST RESULTS FOR W_q IN TERMS OF PAIRED DIFFERENCES

W_q (Average number of jobs in the queue)		Paired Differences					t-Test Value
		Mean	Std. Deviation	Std. Error Mean	50% Confidence Interval of the Difference		
					Lower	Upper	
Pair 1	FCFS - MDJS(2 Server)	-0.008391667	0.012602895	0.007276285	-0.014332728	-0.002450605	-1.153
Pair 1	JS(2 Server) - MDJS(2 Server)	-0.000501333	0.000974385	0.000562562	-0.000960663	-0.000042004	-0.891

TABLE XX
T-TEST RESULTS FOR W_s IN TERMS OF PAIRED DIFFERENCES

W_s (Average number of jobs in the queue)		Paired Differences					t-Test Value
		Mean	Std. Deviation	Std. Error Mean	50% Confidence Interval of the Difference		
					Lower	Upper	
Pair 1	FCFS - MDJS(2 Server)	0.017725000	0.018483767	0.010671608	0.009011669	0.026438331	1.661
Pair 1	JS(2 Server) - MDJS(2 Server)	-0.002165333	0.008847905	0.005108340	-0.006336276	0.002005609	-0.424

IX. CONCLUSION

From the above investigation, the proposed Mid Average Round Robin Scheduling algorithm used in Johnson's sequence compares to the Round Robin Scheduling algorithm by measuring less Service time , Service rate, and reduced number of L_s, L_q, W_q, W_s by implementing them on the given formulas. We conclude that it shows excellent performance. This is achieved by dynamically increasing the amount of time.

REFERENCES

[1] Elaziz, M.A.; Xiong, S.; Jayasena, K.; Li, L. Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowl.-Based Syst. 2019, 169, 39–52.

[2] Malik H. B., Amir M., Mazhar B., Ali S., Jalil R., Khalid J., Comparison of Task Scheduling Algorithms in Cloud Environment,International Journal of Advanced Computer Science and Application, 2018, 9(5), 384-390.

[3] F. Saqib, A. Dutta, J. Plusquellic, P. Ortiz, and M. S. Pattichis, "Pipelined decision tree classifcation accelerator implementation in FPGA (DT-CAIF)," Institute of Electrical and Electronics Engineers. Transactions on Computers, vol. 64, no. 1, pp. 280– 285, 2015

[4] R. Bruni and G. Bianchi, "Efective Classifcation Using a Small Raining Set Based on iscretization and Statistical Analysis," IEEE Transactions On knowledge and data engineering, vol. 27, no. 9, pp. 2349–2361, 2015

[5] Cui, Y., Chin, K. W., Soh, S., Ros, M. (2023). Novel Task Scheduling Approaches in Energy Sharing Solar-Powered IoT Networks. IEEE Internet of Things Journal.

[6] Angiuoli, S. V., Matalka, M., Gussman, A., Galens, K., Vangala, M., Riley, D. R., ... Fricke, W. F. (2011). CloVR: a virtual machine for automated and portable sequence analysis from the desktop using cloud computing. BMC bioinformatics, 12, 1-15.

[7] Singh, Poonam, Maitreyee Dutta, and Naveen Aggarwal. "A review of task scheduling based on meta-heuristics approach in cloud computing." Knowledge and Information Systems 52 (2017): 1-51.

[8] Mansour, R. F., Alhumyani, H., Khalek, S. A., Saeed, R. A., Gupta, D. (2023). Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment. Cluster Computing, 26(1), 575-586.

[9] Waheed, A., Shah, M. A., Mohsin, S. M., Khan, A., Maple, C., Aslam, S., Shamshirband, S. (2022). A comprehensive review of computing paradigms, enabling computation offloading and task execution in vehicular networks. IEEE Access, 10, 3580-3600.

[10] Ibrahim, I. M. (2021). Task scheduling algorithms in cloud computing: A review. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(4), 1041-1053.

[11] Wang, Y., Meng, W., Li, W., Liu, Z., Liu, Y., Xue, H. (2019). Adaptive machine learning-based alarm reduction via edge computing for distributed intrusion detection systems. Concurrency and Computation: Practice and Experience, 31(19), e5101.
[12] Yang, M., Ma, H., Wei, S., Zeng, Y., Chen, Y., Hu, Y. (2020). A multi-objective task scheduling method for fog computing in cyber-physical-social services. IEEE Access, 8, 65085-65095.
[13] Salman, Z., Hammad, M. (2021). Securing cloud computing: A review. International Journal of Computing and Digital Systems, 10, 545-554.
[14] Indukuri, R. K. R., Penmasta, S. V., Sundari, M. R., Moses, G. J. (2016, February). Performance Evaluation of Deadline Aware Multi-Stage Scheduling in Cloud Computing. In 2016 IEEE 6th International Conference on Advanced Computing (IACC) (pp. 229-234). IEEE.
[15] Prasad, A. S., Rao, S. (2013). A mechanism design approach to resource procurement in cloud computing. IEEE Transactions on Computers, 63(1), 17-30.
[16] Karthick, A. V., Ramaraj, E., Subramanian, R. G. (2014, February). An efficient multi queue job scheduling for cloud computing. In 2014 World Congress on Computing and Communication Technologies (pp. 164-166). IEEE.
[17] Zhang, J., Xiong, F., Duan, Z. (2020). Research on resource scheduling of cloud computing based on improved genetic algorithm. Journal of Electronic Research and Application, 4(2).
[18] Luna, J., Taha, A., Trapero, R., Suri, N. (2015). Quantitative reasoning about cloud security using service level agreements. IEEE Transactions on Cloud Computing, 5(3), 457-471.
[19] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.
[20] Banerjee, P., Roy, S. (2021, October). An Investigation of Various Task Allocating Mechanism in Cloud. In 2021 5th International Conference on Information Systems and Computer Networks (ISCON) (pp. 1-6). IEEE.
[21] Fuchigami, H. Y., Rangel, S. (2018). A survey of case studies in production scheduling: Analysis and perspectives. Journal of Computational Science, 25, 425-436.
[22] Komaki, G. M., Kayvanfar, V. (2015). Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. Journal of Computational Science, 8, 109-120.
[23] Yang, Y., Shen, H. (2021). Deep reinforcement learning enhanced greedy optimization for online scheduling of batched tasks in cloud HPC systems. IEEE Transactions on Parallel and Distributed Systems, 33(11), 3003-3014.