

Round Robin Scheduling Algorithm with Improved Dynamic Adaptive

Shashank Kumar¹, Devansh Tiwari¹, Prachi Jha¹

¹PG Student, School of Information Technology & Engineering

Vellore Institute of Technology, Vellore, India

Email: {[shashank.kumar2020](mailto:shashank.kumar2020@vitstudent.ac.in), [devansh.tiwari2020](mailto:devansh.tiwari2020@vitstudent.ac.in), [prachi.jha2020](mailto:prachi.jha2020@vitstudent.ac.in)} @vitstudent.ac.in

ABSTRACT

Round Robin scheduling algorithm is the generally utilized scheduling algorithm in performing various tasks and real time environment. it's the foremost mainstream algorithm because to its reasonableness and starvation free nature towards the processes, which is accomplished by utilizing the time quantum. because the time quantum is static, it causes less context switching if there should arise an event of time quantum and high context switching within the event of less time quantum. Increasing context switch results in high avg. waiting time, high avg. turnaround which is an overhead and degrades the system performance. during this way, the performance of the system exclusively relies on the choice of optimal time quantum which is dynamic in nature. during this paper, we've proposed another variation of RR scheduling algorithm referred to as Improved Round Robin (IRR) Scheduling algorithm, by arranging the processes as per their shortest burst time and allotting all of them with an optimal time quantum which may diminish all the above said inconveniences. Experimentally we've shown that our proposed algorithm performs better than the RR algorithm, by lessening context switching, average waiting and average turnaround.

Watchwords

RR Scheduling, Improved Round Robin (IRR), Context Switching, Average waiting time, Average turnaround time

1 INTRODUCTION

Operating System is an assortment of software modules to assist developers in upgrading system effectiveness and vigor. it's an all-inclusive

machine from the client's perspective and a resource manager from the system perspective. Scheduling is that the most repetitively used fundamental concept in OS. In performing various tasks and multiprogramming environment it's important to select the method among the amount of process present within the job pool consistent with their need. Allocation of CPU to the processes is completed by scheduler, which operated by some scheduling algorithms. FCFS, SJF, Priority & RR are different sort of scheduling algorithms. during which RR is that the hottest non-preemptive scheduling algorithm. In non- preemption, CPU is assigned to a process until its execution is completed. But in preemption, running process is forced to release the CPU by the newly arrived process. Each scheduling algorithm has its own advantages and drawbacks. Similarly, RR features a drawback which increase average waiting time, average turnaround and minimizes the throughput, referred to as Context switch. The processes in RR are assigned with a time quantum which is static naturally.

1.1 MOTIVATION

In RR scheduling, processes get justifiable share of CPU due to static time quantum assign to every process and therefore the context switch is inversely proportional too choice of static time quantum which degrades the general performance of the system (high average waiting time & average turnaround time). This factor motivates us to style an improved algorithm which is in a position to extend the system performance by reducing the amount of context switches, average waiting time & average turnaround using the concept of dynamic time quantum.

1.2 FOCUS

Aims and objective of research is to schedule the processes in efficient and convenient way. Scheduling decision attempt to reduce the following: turnaround, reaction time and average waiting time for processes and therefore the number of context switches. therefore, the main objective of research to scale back the load of CPU work, increase the performance of CPU, reduces the overhead and schedule the task in efficient manner. during this paper, we present an improved version of Round Robin with varying time quantum. the most contributions of our work are often summarized as follows:

1. Proposes an analytical model that takes into consideration different parameters to work out the order of tasks execution. the utilization of burst time as parameters in our model ensures a more suitable time quantum.
2. Reduce the typical turnaround ATAT and therefore the average waiting time AWT of existing tasks, which results in a far better use of resources that's considered together of the main problems within the OS and cloud computing environment.

2. RESEARCH GAP IDENTIFIED

However, tons of algorithms are introduced over past few decades for reduce or improve the performance of system but most of the prevailing models of scheduling are fails in real time-frame. To fill this gap a replacement endeavor has been made during this paper to enhance the performance of scheduling. The designed scheduling approach has simulated and analysis with two hottest algorithm IRR Scheduling Algorithm & RR. The results demonstrated that presented approach presents efficient outcomes over the normal scheduling algorithms.

PROPOSED APPROACH

In our proposed algorithm, we are arranging the processes in ascending order consistent with their burst time present within the ready queue. for locating an optimal time quantum, median method is followed. The median are often calculated using the subsequent formulae.

$$\text{Median}(M) = \begin{cases} Y_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \\ \frac{Y_n + Y_{n+1}}{2} & \text{if } n \text{ is even} \end{cases}$$

Where, M = median

Y = number located within the middle of a group of numbers arranged in ascending order

n = number of processes

Then, the optimal time quantum is calculated as follows:

$$\text{Optimal Time Quantum (Oqt)} = \frac{\text{HighestBt} + \text{Median}(M)}{2}$$

The optimal time quantum is assigned to each process and is recalculated taking the remaining burst time in account after each cycle. This procedure goes on until the ready queue is empty.

2.1 Proposed Algorithm

1. I/P: Process (Pn), Burst Time(bt), Arrival Time, ready queue. O/P: Context Switch (Cs), Average waiting-time (Awt), Avg.Turnarround Time(Att)
2. Initialize: ready queue=0, Cs=0, Awt=0, Att=0
3. While (ready queue!=0)
 - // Sort the processes in ascending order according to their Bt in ready queue
 - //Find Median
 - //Calculate Oqt
 - 1. //assign Oqt to each process for each process i=1 to n P[i]
Oqt
 - 2. If new process arrives
Update counter and
goto step3 end while
 - 3. Cs, Awt, Att are calculated.
 - 4. Stop and exit.

2.2 Illustration

To demonstrate the above algorithm, we've considered the subsequent example. time of arrival is taken into account to be zero for the given processes P1, P2, P3, P4 and corresponding burst times are 60, 20, 80, 40 respectively. In initiative

the processes within the ready queue are sorted in ascending order. Then the time quantum is calculated within the second step. Here $Oqt = 65$. In third step sorted processes are executed with the optimal time quantum, i.e, P2 with $bt = 20$, P4 having $bt=40$, P1 with $bt=60$ and P3 with $bt=80$. After assigning Oqt to every process the remaining burst time of all process are $P1=0$, $P2=0$, $P3=15$ and $P4=0$. When a process completes its execution, it's deleted from ready queue automatically. Then subsequent time quantum is calculated from remaining burst times as per the 3rd step within the algorithm. Here $Oqt=15$. Then the remaining burst times are $P3=0$. In the last step P3 will complete its execution and can be deleted from the ready queue.

3. EXPERIMENTAL ANALYSIS

3.1 Assumptions

Assumptions All the processes are assumed to be independent. Time slice is assumed to be less than the utmost burst time. All the attributes like burst time, number of processes and therefore the time slice of all the processes are known before submitting the processes to the processor. All processes are CPU bound. No processes are I/O bound.

3.2 Experimental Frame Work

Our experiment consists of several input and output parameters. The input parameters contain burst time (Bt), time of arrival (At), optimal time quantum (Oqt) and therefore the number of processes (Pn). The output parameters contain average waiting time (Awt), average turnaround (Att) and number of context switches (Cs).

3.3 Result Obtained

Obtained Our proposed algorithm can work effectively with sizable amount of knowledge. In each case we've compared our proposed algorithm's results with Round Robin scheduling algorithm's result. For RR Scheduling Algorithm we've taken 25 because the static time quantum.

Case 1: With Zero Arrival Time

Increasing Order

We consider five processes P1, P2, P3, P4 and P5 arriving at time 0 with burst time 30, 34, 62, 74, 88 respectively shown in Table 3.1. Table 3.2 shows

the comparing result of RR algorithm and our proposed algorithm.

No. of process	At	Bt
P1	0	30
P2	0	34
P3	0	62
P4	0	74
P5	0	88

Table 1. Data in Increasing Order

Algorithm	RR	IRR
qt	25	75,13
Cs	13	5
Awt	149	85.6
Att	206.6	143.2

Table 2. Comparison between RR and IRR

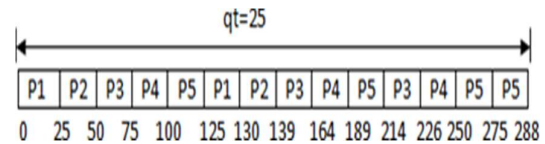


Fig 1: Gantt chart for RR in Table2

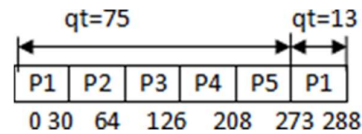


Fig 2: Gantt chart for IRR in Table 2

Decreasing Order

We consider five processes P1, P2, P3, P4 and P5 arriving at time 0 with burst time 77, 54, 45, 19, 14 respectively shown in Table 3. Table 4 shows the comparing result of RR algorithm and our proposed algorithm.

Table 3. Data in Decreasing Order

No. of process	At	Bt
P1	0	77
P2	0	54
P3	0	45
P4	0	19
P5	0	14

Table 4. Comparison between RR and IRR

Algorithm	RR	IRR
qt	25	61,16
Cs	10	5
Awt	117.4	51.4
Att	159.2	93.2

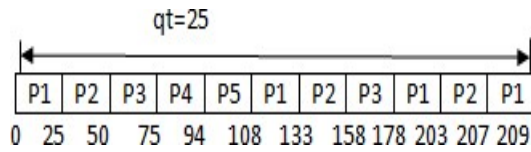


Fig 3: Gantt chart for RR in Table 4

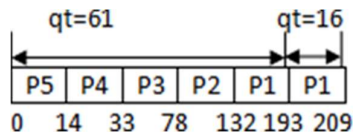


Fig 4: Gantt chart for IRR in Table 4

Random Order

We consider five processes P1, P2, P3, P4 and P5 arriving at time 0 with burst time 80, 45, 62, 34, 78 respectively shown in Table 3.5. Table 3.6 shows the comparing result of RR algorithm and our proposed algorithm

Table 5. Data in Random Order

No. of process	At	Bt
P1	0	80
P2	0	45
P3	0	62
P4	0	34
P5	0	78

Table 6. Comparison between RR and IRR

Algorithm	RR	IRR
qt	25	71,8,1
Cs	14	7
Awt	187.2	108.8
Att	247	168

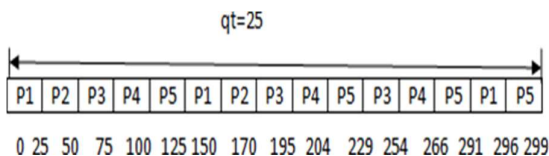


Fig 5: Gantt chart for RR in Table 6

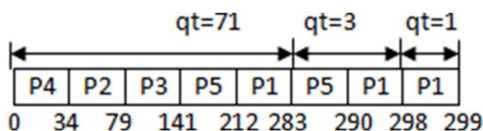


Fig 6: Gantt chart for IRR in Table 6

Case 2: Without Zero

Arrival Time Increasing

Order

We consider five processes P1, P2, P3, P4 and P5 arriving at time 0, 2, 6, 8, 14 and burst time 14, 34, 45, 62, 77 respectively shown in Table 3.7. Table 3.8 shows the comparing result of RR algorithm and our proposed algorithm.

Table 7. Data in Increasing Order

No. of process	At	Bt
P1	0	14
P2	2	34
P3	6	45
P4	8	62
P5	14	77

Table 8. Comparison between RR and IRR

Algorithm	RR	IRR
qt	25	14,65,12
Cs	11	5
Awt	97	56.6
Att	143.4	103

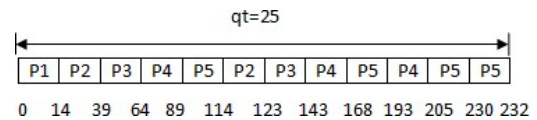


Fig 7: Gantt chart for RR in Table 8

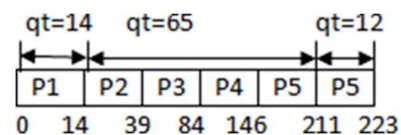


Fig 8: Gantt chart for IRR in Table 8

Decreasing Order

We consider five processes P1, P2, P3, P4, P5 arriving at time 0, 2, 3, 4, 5 and burst time 80, 74, 70, 18, 14 respectively shown in Table 3.9. Table 3.10 shows the comparing result of RR algorithm and our proposed algorithm.

Table 9. Data in Decreasing Order

No. of process	At	Bt
P1	0	80
P2	2	74
P3	3	70
P4	4	18
P5	5	14

Table 10. Comparison between

RR and IRR

Algorithm	RR	IRR
qt	25	80,59,14,1
Cs	11	7
Awt	136.4	105.4
Att	187.6	156.6

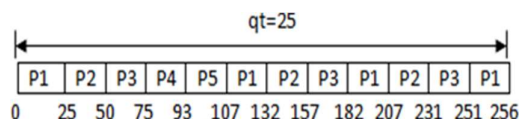


Fig 9: Gantt chart for RR in Table 10

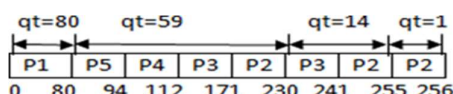


Fig 10: Gantt chart for RR in Table 10

Random Order

We consider five processes P1, P2, P3, P4 & P5 arriving at time 0, 1,4,6,7 and burst time 65,72,50,43,80 respectively shown in Table 3.11. Table 3.12 shows the comparing result of RR algorithm and our proposed algorithm.

Table 11. Data in Random Order

No. of process	At	Bt
P1	0	65
P2	1	72
P3	4	50
P4	6	43
P5	7	80

Table 12. Comparison between RR and IRR

Algorithm	RR	IRR
qt	25	72,8
Cs	13	5
Awt	199.8	104.8
Att	261.8	166.8

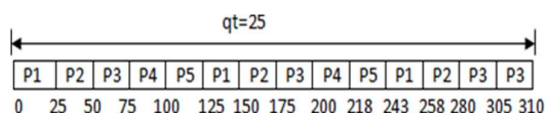


Fig 11: Gantt chart for RR in Table 12

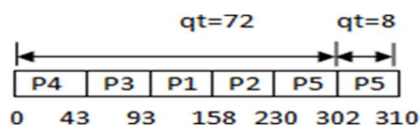


Fig 12: Gantt chart for IRR in Table 12

4. CONCLUSION AND FUTURE WORK

From the above experimental analysis, it is proved that our proposed IRR scheduling algorithm gives better result than classical RR scheduling algorithm by taking avg. waiting time, avg. turnaround time and context switch as performance parameter. Our proposed algorithm can be extended using soft and hard real time systems.

5. LITERATURE SURVEY AND REFERENCES

- [1] Sukumar Babu Bandrupalli, Neelima Priyanka Nutulapati and Prof. Dr. P. Suresh Varma "A Novel CPU Scheduling Algorithm–Preemptive & Non-Preemptive" International Journal of Modern Engineering Research (IJMER) Vol.2, Issue.6, Nov-Dec. 2012 pp-4484-4490. ["The main objective of this paper is to introduce a new CPU algorithm called A Novel CPU Scheduling Algorithm which acts as both preemptive and non-preemptive based on the arrival time. The proposed algorithm helps to improve the CPU efficiency in real time uni-processor-multi programming operating system. CPU Scheduling is the basis of multi-programmed operating system. The scheduler is responsible for multiplexing processes on the CPU. There are many scheduling algorithms available for a multi-programmed operating system like FCFS, SJF, Priority, Round Robin etc. In this paper, the results of the existing algorithms (FCFS, SJF, Priority and Round Robin) are compared with the proposed algorithm."]
- [2] Ajit Singh, Priyanka Goyal and Sahil Batra "An Optimized Round Robin Scheduling Algorithm for CPU Scheduling" International Journal on Computer Science and Engineering Vol. 02, No. 07, 2010, 2383-2385. ["The main objective of this paper is to develop a new approach for round robin scheduling which help to improve the CPU efficiency in real time- and time algorithms available for CPU scheduling. But we cannot implement in real time operating system because of high context switch rates, large waiting time, large response time, large turn around time and less throughput. The proposed algorithm improves all the drawback of simple round robin architecture. The author has also given comparative analysis of

proposed with simple round robin scheduling algorithm. Therefore, the author strongly feel that the proposed architecture solves all the problem encountered in simple round robin architecture by decreasing the performance parameters to desirable extent and thereby increasing the system throughput.”]

- [3] Manish Chandra Bhatt and Nikhil Kumar Singh “An Analytical Study of CPU Scheduling Algorithms” International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 7, July 2016. [“Present paper is the study about most of the CPU scheduling algorithms and its features. In this paper the comparison between the algorithms on the same CPU is shown. Using this comparison one can easily understand that what is performing inside the CPU. The aim of this survey is to analyze that CPU scheduler which have maximum efficiency and may also satisfy all the objectives of the scheduling.”]
- [4] Alaa Fiad, Zoulikha Mekakia Maaza and Hayat Bendoukha “Improved Version of Round Robin Scheduling Algorithm Based on Analytic Model” International Journal of Networked and Distributed Computing Vol. 8(4); December (2020), pp. 195–202. [“Scheduling is one of the most important issues in the operating system, such as the tasks must be affected to the appropriate virtual machines, considering different factors at the same time to ensure better use of resources. A lot of research has been carried out to propose more efficient task scheduling algorithms. Round robin is one of the most powerful algorithms in this field but its main challenge is the choice of time quantum. The effectiveness of Round Robin depends on the choice of this parameter. In this paper, we aimed to overcome this challenge by proposing an improved Round Robin scheduling algorithm using a variable time quantum based on an analytic model. Our analytical model takes into consideration different parameters to determine the order of tasks execution. The use of burst time as parameters in our model ensures a more suitable time quantum. This algorithm can be applied in any operating system and therefore in the cloud computing environment. In order to assess the performance and effectiveness of the proposed approach, five different scenarios have been implemented; the comparisons made with existing works have shown that the proposed approach improves the average waiting time and the average turnaround time, which ensures better scheduling of tasks and better use of resources.”]
- [5] Khazi Faizan Ali, Abhijeet Marikal, Kakelli Anil Kumar “A Hybrid Round Robin Scheduling Mechanism for Process Management” International Journal of Computer Applications (0975 – 8887) Volume 177 – No. 36, February 202. [“Every process which runs in an operating system is processed through multiprogramming, by sharing the processor. Multiprogramming is a capability of the operating system to execute one or a lot of programs in a single processing machine. Multiprogramming's main objective is to optimize the use of CPUs, throughput and minimize average waiting time (AWT), the average turnaround time (ATT), average response time (ART) and the number of context switches (NOC). Multi-programming can be performed by several CPU scheduling algorithms. Most preferable scheduling algorithm among all is Round Robin (RR). RR algorithm is mainly depended on time quantum which is stationary. Hence, the proposed work is new approach to the round-robin scheduling algorithm (RR algorithm) with an enhanced time quantum-based algorithm. This enhancement of RR algorithm by using dynamic time quantum leads to minimize AWT, ATT, ART and NOC. This approach inherits the properties of Round Robin, shortest job first (SJF) algorithm and first come first serve algorithm (FCFS). Therefore, the proposed algorithm is a hybrid round robin scheduling mechanism for process management (HYRR Mechanism). HYRR Mechanism is an innovative scheduling algorithm which reduces ATT, ART, AWT and NOC to the desired levels.”]
- [6] Ali A. AL-Bakhrani, Abdulnaser A. Hagar, Ahmed A. Hamoud and Seema Kawathekar “Comparative Analysis of Cpu Scheduling Algorithms: Simulation and Its Applications” International Journal of Advanced Science and Technology Vol. 29, No. 3, (2020), pp. 483-494. [“In this paper, we seek to present the services provided by the simulation program, which can be used in the educational aspect as

it facilitates the student to understand the scheduling algorithms within the CPU.”]

- [7] Saroj Hiranwal and K. C. Roy “Adaptive Round Robin Scheduling Using Shortest Burst Approach Based on Smart Time Slice” International Journal of Computer Science and Communication Vol. 2, No. 2, July- December 2011, pp. 319-323.
- [8] Operating Systems ,3rd Ed., H.M. Deitel, P.J. Deitel, D.R. Choffnes. ISBN 978-81-317-1289-4.
- [9] Operating System Concepts ,8th Ed., Abraham Silber Schatz, Peter B. Galvin, Grege Gagne. ISBN 978- 81-265-2051-0.
- [10] Prof. Rakesh Mohanty, Manas Das, M. Lakshmi Prasanna and Sudhashree “Design and Performance Evaluation of a New Proposed Fittest Job First Dynamic Round Robin (FJFDRR) Scheduling Algorithm” International Journal of Computer Information Systems, Vol. 2, No. 2, 2011. [“ In this paper, we have proposed a new variant of Round Robin scheduling algorithm by executing the processes according to the new calculated Fit Factor ‘f’ and using the concept of dynamic time quantum. We have compared the performance of our proposed Fittest Job First Dynamic Round Robin (FJFDRR) algorithm with the Priority Based Static Round Robin (PBSRR) algorithm. Experimental results show that our proposed algorithm performs better than PBSRR in terms of reducing the number of context switches, average waiting time and average turnaround time.”]
- [11] Prof. Rakesh Mohanty, Prof. H. S. Behera Khusbu Patwari, Manas Ranjan Das, Monisha Dash and Sudhashree “Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm”. [“In this paper, we have studied various algorithms for scheduling of the processes in Operating System (OS). Amongst all the algorithms, Round Robin (RR) performs optimally in timeshared system because each process is given an equal amount of static time quantum. But the effectiveness of RR algorithm solely depends upon the choice of time quantum. We

have made a comprehensive study and analysis of RR algorithm. We have proposed a new Improved-RR algorithm SRBRR (Shortest Remaining Burst Round Robin) by assigning the processor to processes with shortest remaining burst in round robin manner using the dynamic time quantum. Our experimental analysis shows that SRBRR performs better than RR algorithm in terms of reducing the number of context switches, average waiting time and average turnaround time.”]

- [12] H.S. Behera, R. Mohanty and Debashree Nayak “A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis” International Journal of Computer Applications, Vol. 5, No. 5, August 2010. [“Scheduling is the central concept used frequently in Operating System. It helps in choosing the processes for execution. Round Robin (RR) is one of the most widely used CPU scheduling algorithm. But its performance degrades with respect to context switching, which is an overhead and it occurs during each scheduling. Overall performance of the system depends on choice of an optimal time quantum, so that context switching can be reduced. In this paper, we have proposed a new variant of RR scheduling algorithm, known as Dynamic Quantum with Readjusted Round Robin (DQRRR) algorithm. We have experimentally shown that performance of DQRRR is better than RR by reducing number of contexts switching, average waiting time and average turnaround time.”]
- [13] H. S. Behera, Rakesh Mohanty, Sabyasachi Sahu and Sourav Kumar Bhoi “COMPARATIVE PERFORMANCE ANALYSIS OF MULTI-DYNAMIC TIME QUANTUM ROUND ROBIN (MDTQRR) ALGORITHM WITH ARRIVAL TIME” H. S. Behera et al. / Indian Journal of Computer Science and Engineering (IJCSE). [“CPU being considered a primary computer resource; its scheduling is central to operating-system design. A thorough performance evaluation of various scheduling algorithms manifests that Round Robin Algorithm is considered as optimal in time shared environment because the static time is equally shared among the processes. We have proposed an efficient technique in the process scheduling algorithm by using dynamic time quantum in Round

Robin. Our approach is based on the calculation of time quantum twice in single round robin cycle. Taking into consideration the arrival time, we implement the algorithm. Experimental analysis shows better performance of this improved algorithm over the Round Robin algorithm and the Shortest Remaining Burst Round Robin algorithm. It minimizes the overall number of context switches, average waiting time and average turn-around time. Consequently, the throughput and CPU utilization is better.”]

- [14] Rakesh K. Lenka and Prabhat Ranjan “A 2LFQ Scheduling with Dynamic Time Quantum using Mean Average” International Journal of Computer Applications (0975 – 8887) Volume 47– No.23, June 2012. [“The efficiency and performance of multitasking operating systems essentially depends on the nature of CPU scheduling algorithm. There are many algorithms available for CPU scheduling. Each having its own deficiency and limitations. One of the most well-known approaches for scheduling is the Multi-level Feedback Queue (MLFQ). The MLFQ tries to work in a two-fold manner. First, it tries to optimize turnaround time as it is done by running shorter jobs first. Unfortunately, the OS doesn’t generally have the knowledge that how long a job will run for, exactly the knowledge that algorithms like SJF (or SRTF) require. Second, MLFQ attempts to make a system feel responsive to interactive users (i.e., users sitting and staring at the screen, waiting for a process to finish), and thus minimize response time. Wellknown algorithms like Round Robin also reduce response time but are less suitable for turnaround time. In this paper, we proposed a new approach for feedback scheduling algorithm which helps to improve the efficiency of CPU. The paper presents an approach called dynamic-time-quantum 2LFQ (Two-level Feedback Queue) scheduling. The idea is to make the operating systems adjust the time quantum according to the burst time of set of waiting processes in the ready queue.”]

- [15] Sourav Kumar Bhoi, Sanjaya Kumar Panda and Debashee Tarai “ENHANCING CPU PERFORMANCE USING SUBCONTRARY MEAN DYNAMIC ROUND ROBIN (SMDRR) SCHEDULING ALGORITHM” Volume 2, No. 12, December 2011 Journal of

Global Research in Computer Science. [“Round Robin (RR) Algorithm is considered as optimal in time shared environment because the static time is equally shared among the processes. If the time quantum taken is static then it undergoes degradation of the CPU performance and leads to so many context switches. In this paper, we have proposed a new effective dynamic RR algorithm SMDRR (Subcontrary Mean Dynamic Round Robin) based on dynamic time quantum where we use the subcontrary mean or harmonic mean to find the time quantum. The idea of this approach is to make the time quantum repeatedly adjusted according to the burst time of the currently running processes. Our experimental analysis shows that SMDRR performs better than RR algorithm in terms of reducing the number of context switches, average turnaround time and average waiting time.”]

- [16] Vishnu Kumar Dhakad and Lokesh Sharma “PERFORMANCE ANALYSIS OF ROUND ROBIN SCHEDULING USING ADAPTIVE APPROACH BASED ON SMART TIME SLICE AND COMPARISON WITH SRR” International Journal of Advances in Engineering & Technology, May 2012.[“The Time Sharing System is more multifaceted about the performance and calculating the average waiting time, turnaround time, response time, and context switches from the number of processes mainly depends on the CPU scheduling algorithm where the CPU is one of the most important computer resource and as round robin scheduling is considered most widely used scheduling algorithms. The Round Robin Scheduling Algorithm has disadvantage is longer average waiting time, higher context switches, higher turnaround time and low throughput. In this research a new proposed algorithm called Adaptive Round Robin Scheduling Algorithm is presented which is discussed here. The new proposed algorithm called “Adaptive Round Robin (ARR) Scheduling a Novel Approach Based on Shortest Burst Time Using Smart Time Slice”. It is a Priority Driven Scheduling Algorithm based on burst time of processes. It is also using fixed time quantum same as Simple RR but calculation of time quantum is another factor here. First of all, we arrange the processes according to the execution time/burst time in increasing order that is smallest the burst time higher the priority of the running process. The

next idea of this approach is to choose the Smart Time Slice (STC) is mainly depends on number of processes. The Smart Time Slice is equal to the mid process burst time of all CPU burst time when number of process given odd. If number of process given even then we choose the time quantum according to the average CPU burst of all running processes. The use of scheduling algorithm increased the performance and stability of the time-sharing systems and support building of a self-adaptation operating system, which means that the system is who will adapt itself to the requirements of the user and vice versa.”]

greater number of context switches, larger waiting time and larger response time. We have proposed a novel algorithm, known as Priority Based Dynamic Round Robin Algorithm (PBDRR), which calculates intelligent time slice for individual processes and changes after every round of execution. The proposed scheduling algorithm is developed by taking dynamic time quantum concept into account. Our experimental results show that our proposed algorithm performs better than algorithm in [8] in terms of reducing the number of context switches, average waiting time and average turnaround time.”]

[17] Ms. Rukhsar Khan and Mr. Gaurav Kakhani “Analysis of Priority Scheduling Algorithm on the Basis of FCFS & SJF for Similar Priority Jobs” IJCSMC, Vol. 4, Issue. 9, September 2015, pg.324 – 331. [“Scheduling is one of the most important activity of the process manager which take decision to choose which of the process in the ready queue will be assigned to the CPU. There are different types of scheduling algorithms available for taking decision. One of them is Priority Scheduling Algorithm, which is based on the priority assigned to each process. In priority scheduling the Processes are executed on the basis of priority, the process having highest priority is executed first. In case of similar priority FCFS is used. In this paper, the priority scheduling algorithm is used in such a way that, in case of similar priority SJF algorithm is used instead of FCFS and average waiting time and average turnaround time is calculated. The comparative analysis is performed on the SJF based priority scheduling and FCFS based priority scheduling to compare the average waiting time and average turnaround time.”]

[19] Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash and Himanshu Sharma “An Improved Round Robin Scheduling Algorithm for CPU scheduling” International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1064-1066. [“There are many functions which are provided by operating system like process management, memory management, file management, input/output management, networking, protection system and command interpreter system. In these functions, the process management is most important function because operating system is a system program that means at the runtime process interact with hardware. Therefore, we can say that for improving the efficiency of a CPU we need to manage all process. For managing the process, we use various types scheduling algorithm. There are many algorithms are available for CPU scheduling. But all algorithms have its own deficiency and limitations. In this paper, I proposed a new approach for round robin scheduling algorithm which helps to improve the efficiency of CPU.”]

[18] Prof. Rakesh Mohanty, Prof. H. S. Behera, Khusbu Patwari, Monisha Dash and M. Lakshmi Prasanna “Priority Based Dynamic Round Robin (PBDRR) Algorithm with Intelligent Time Slice for Soft Real Time Systems” (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.2, February 2011. [“In this paper, a new variant of Round Robin (RR) algorithm is proposed which is suitable for soft real time systems. RR algorithm performs optimally in timeshared systems, but it is not suitable for soft real time systems. Because it gives a

[20] Debashree Nayak, Sanjeev Kumar Malla and Debashree Debadarshini “Improved Round Robin Scheduling using Dynamic Time Quantum” International Journal of Computer Applications (0975 – 8887) Volume 38– No.5, January 2012. [“Round Robin scheduling algorithm is the widely used scheduling algorithm in multitasking and real time environment. It is the most popular algorithm due to its fairness and starvation free nature towards the processes, which is achieved by using the time quantum. As the time quantum is static, it causes less context switching in case of high time quantum and high context

switching in case of less time quantum. Increasing context switch leads to high avg. waiting time, high avg. turnaround time which is an overhead and degrades the system performance. So, the performance of the system solely depends upon the choice of optimal time quantum which is dynamic in nature. In this paper, we have proposed a new variant of RR scheduling algorithm known as Improved Round Robin (IRR) Scheduling algorithm, by arranging the processes according to their shortest burst time and assigning each of them with an optimal time quantum which is able to reduce all the above said disadvantages. Experimentally we have shown that our proposed algorithm performs better than the RR algorithm, by reducing context switching, average waiting and average turnaround time.”]

[21] S. R. Chavan and P. C. Tikekar “An Improved Optimum Multilevel Dynamic Round Robin Scheduling Algorithm” International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013 ISSN 2229-5518. [“The main objective of this paper is to improve the Round Robin scheduling algorithm using the dynamic time slice concept. One of the fundamental functions of an operating system is scheduling. Sharing of computer resources between multiple processes is called as scheduling. The main use of round robin scheduling algorithms is in the real time systems but still it has several difficulties The intention should be allowed as many as possible running processes at all time in order to make best use of CPU. CPU scheduling has strong effect on resource utilization as well as overall performance of the system. Round Robin algorithm performs optimally in timeshared systems, but it is not suitable for soft real time systems, because it gives a greater number of context switches, larger waiting time and larger response time. In this paper, a new CPU scheduling algorithm called An Optimum Multilevel Dynamic Round Robin Scheduling Algorithm is proposed, which calculates intelligent time slice and changes after every round of execution. The suggested algorithm was evaluated on some CPU scheduling objectives and it was observed that this algorithm gave good performance as compared to the other existing CPU scheduling algorithms.”]

[22] MEHDI NESHAT, MEHDI SARGOLZAEI, ADEL NAJARAN and ALI ADELI “THE NEW METHOD OF ADAPTIVE CPU SCHEDULING USING FONSECA AND FLEMING’S GENETIC ALGORITHM” Journal of Theoretical and Applied Information Technology 15th March 2012. Vol. 37 No.1. [“The CPU scheduling is one of the most important tasks of the operating system. Many algorithms are designed and used in this regard each having advantages and disadvantages. In this paper a new algorithm for the CPU scheduling is presented using FFGA (Fonseca and Fleming’s Genetic Algorithm) multiobjective optimization. Contrary to the classical algorithms in use, it uses the three parameters of CPU burst time; I/O devices service time, and priority of process instead of using one parameter of CPU burst time. The important point is the adaptation of the algorithm which selects a special process depending on the system situation. The performance of this algorithm was compared with seven classical scheduling algorithms (FCFS, RR (equal, prioritized), SJF (preemptive, non-preemptive, Priority (preemptive, nonpreemptive)), and the results showed that the performance of the proposed method is more optimized than other methods. The proposed algorithm optimizes the average waiting time and response time for the processes.”]

[23] R.I. Davis and A. Burns “An Investigation into Server Parameter Selection for Hierarchical Fixed Priority Pre-emptive Systems”. [“This paper investigates the problem of server parameter selection in hierarchical fixed priority preemptive systems. A set of algorithms are provided that determine the optimal values for a single server parameter (capacity, period, or priority) when the other two parameters are fixed. By contrast, the general problem of server parameter selection is shown to be a holistic one: typically, the locally optimal solution for a single server does not form part of the globally optimal solution.”]

[24] Neetu Goel and Dr. R.B. Garg “A Comparative Study of CPU Scheduling Algorithms”. [“This paper presents a state diagram that depicts the comparative study of various scheduling algorithms for a single CPU and shows which algorithm is best for the particular situation.

Using this representation, it becomes much easier to understand what is going on inside the system and why a different set of processes is a candidate for the allocation of the CPU at different time. The objective of the study is to analyze the highly efficient CPU scheduler on design of the high-quality scheduling algorithms which suits the scheduling goals.”]

[25] Jyotirmay Patel and A.K. Solanki “Performance Evaluation of CPU Scheduling by Using Hybrid Approach” International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 4, June – 2012. [“Central Processing Unit (CPU) scheduling plays crucial role by switching the CPU among various processes. The problem of scheduling which computer process run at what time on the central processing unit (CPU) or the processor is explored. Some CPU scheduling algorithms has been elaborated and assessed on the basic CPU scheduling objectives i.e.; average waiting time, turnaround time etc. These will form the base parameters in making a decision for the suitability of the given algorithm for a given objective. Many algorithms have been developed for the CPU scheduling of a modern multiprogramming operating system. Our research work involves the design and development of new CPU scheduling algorithm (the Hybrid Scheduling Algorithm). This work involves a software tool which produces a comprehensive simulation of a number of CPU scheduling algorithms. The tool’s results are in the form of scheduling performance metrics.”]

[26] G. Siva Nageswara Rao, N. Srinivasu, S.V.N. Srinivasu and G. Rama Koteswara Rao “Dynamic Time Slice Calculation for Round Robin Process Scheduling Using NOC” International Journal of Electrical and Computer Engineering (IJECE) Vol. 5, No. 6, December 2015, pp. 1480~1485. [“Process scheduling means allocating a certain amount of CPU time to each of the user processes. One of the popular scheduling algorithms is the “Round Robin” algorithm, which allows each and every process to utilize the CPU for short time duration. This paper presents an improvisation to the traditional round robin scheduling algorithm, by the proposed a new method. The new method represents the time slice as a function of the burst time of the

waiting process in the ready queue. Fixing the time slice for a process is a crucial factor, because it subsequently influences many performance parameters like turnaround time, waiting time, response time and the frequency of context switches. Though the time slot is fixed for each process, this paper explores the fine-tuning of the time slice for processes which do not complete in the stipulated time allotted to them.”]

[27] H.S. Behera, Brajendra Kumar Swain, Anmol Kumar Parida and Gangadhar Sahu “A New Proposed Round Robin with Highest Response Ratio Next (RRHRRN) Scheduling Algorithm for Soft Real Time Systems” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1 Issue-3, February 2012. [“ The efficiency and performance of multitasking operating systems mainly depend upon the use of CPU scheduling algorithms. Round Robin (RR) performs optimally in timeshared system but it is not suitable for real time system because it gives a greater number of context switches, larger waiting and turnaround time. In this paper, we have proposed a new Round Robin with Highest Response Ratio Next (RRHRRN) scheduling algorithm, which uses Highest Response Ratio (HRR) criteria for selecting processes from Ready Queue. Our experimental result shows that our proposed algorithm performs better than algorithm in DQRRR [1] in terms of reducing the number of context switches, average waiting time and average turnaround time.”]

[28] Pallab Banerjee, Probal Banerjee and Shweta Sonali Dhal “Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-1, Issue-3, August 2012.[“ Round Robin Scheduling algorithm is designed especially for time sharing Operating system (OS).It is a preemptive CPU scheduling algorithm which switches between the processes when static time Quantum expires. The Round Robin Scheduling algorithm has its disadvantages that is its longer average waiting time, higher context switches, higher turnaround time. In this paper a new algorithm is presented called Average Max Round Robin (AMRR) scheduling algorithm. In this scheduling algorithm the main idea is to adjust the time Quantum dynamically so that

(AMRR) perform better performance than simple Round Robin scheduling algorithm.”]

- [29] Pooja Samal and Pranati Mishra “Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing”) International Journal of Computer Science and Information Technologies, Vol. 4 (3), 2013, 416-419. [“Cloud computing is the emerging internet-based technology which emphasizes commercial computing. Cloud is a platform providing dynamic pool resources and virtualization. Based on a pay-as-you-go model, it enables hosting of pervasive applications from consumer, scientific, and business domains. To properly manage the resources of the service provider, load balancing is required for the jobs that are submitted to the service provider. Load balancing also helps in improving the performance of the centralized server. In the present work, various policies in relation to the algorithms developed are analyzed using an analysis tool, namely, cloud analyst. Comparison is also made for variants of Round Robin (RR) algorithms.”]
- [30] Rakesh Patel and Mrs. Mili Patel “SJRR CPU Scheduling Algorithm” International Journal of Engineering and Computer Science ISSN: 2319-7242, Volume 2, Issue 12, Dec.2013. [“The main objective of this paper is to introduce a new CPU algorithm called SJRR CPU Scheduling Algorithm which acts as preemptive based on the arrival time. The algorithm helps to improve the average waiting time of Round Robin algorithm in real time uni-processor-multi programming operating system. CPU Scheduling is the basis of multi-programmed operating system. The scheduler is responsible for multiplexing processes on the CPU. There are many scheduling algorithms available for a multi-programmed operating system like FCFS, SJF, Priority, Round Robin etc. The proposed algorithm is based on Round Robin scheduling. In this paper, the results of the existing Round Robin algorithm are compared with the proposed algorithm.”]