A Comparative Analysis of Quantum Time based on Pythagorean Means for Dynamic Round Robin Scheduling

ABSTRACT

Round-robin scheduling (RRS) is a job-scheduling algorithm that is considered to be fair, for it handles the request in a circular first-in-first-out (FIFO) order and is considered as a cycle executive. Due to these processes, it has a fair and starvation free nature, which is achieved by using proper quantum time. Quantum time is fixed in most round-robin and processes are scheduled such that no process gets more than one-time quantum in one go. Due to this, the algorithms should calculate the proper value for quantum time. In this paper, we compare and analyze different means of computing for the quantum time in each cycle. The first method is Arithmetic Mean which is simply taking the sum of all data and then dividing the sum by the total number of values, it is commonly referred to as an average. The second method is using Geometric Mean which indicates the central tendency or typical value of a set of numbers by using the product of their values. The last method is Harmonic Mean which is calculated by dividing the number of observations by the reciprocal of each number in the series. These methods will provide a better scheduling criterion for the average Turnaround time, average waiting time, response time and context switches. Amongst these methods, Harmonic Arithmetic Mean proves to be more efficient in computing the Quantum time.

INTRODUCTION

CPU scheduling plays a vital role that is central to designing an operating system. There are cases where there is more than one process in the queue waiting to be assigned to the CPU, the operating system must determine what to prioritize using the scheduler for the order of execution. Allocating CPU to a process requires careful attention to assure fairness and avoid process starvation this can have a big effect on resource utilization and other performance parameters [1]. Operating systems aim to allow several processes concurrently to maximize CPU utilization [2].

This paper aims to compare and analyze an existing Round-Robin (RR) scheduling algorithm based on Arithmetic Mean, Geometric Mean, and Harmonic Mean. The RR algorithm is one of the oldest, simplest and fairest and most widely used scheduling algorithms, designed especially for time-sharing systems [3]. Processes are dispatched in a first-in-first-out sequence but each process is allowed to run for only a limited amount of time. This time interval is known as a time-slice or quantum time which can be more efficient if it is computed and implemented properly [4]. The algorithms implemented using quantum time factor to compare which mean is the most efficient scheduling criteria that decrease the average turnaround time, average waiting time, average response time and context switching.

REVIEW OF RELATED LITERATURE

CPU optimization of scheduling algorithms is based on the average value of criteria by either maximizing the CPU utilization and throughput or minimizing the all. There are several different criteria to consider when trying to select the ideal scheduling algorithm for a particular situation and environment. The criteria include the following:

CPU utilization - Preferably the CPU is busy 100% of the time, which is to waste 0 CPU cycles. The range of CPU usage should be from 40% which is lightly loaded to 90% that is heavily loaded on a real system.

Throughput - Number of processes completed per unit time that may range from 10 seconds to 1 hour depending on the specific processes.

Turnaround time - Also known as Wall Clock Time that is required for a particular process to complete, from submission time to completion.

Waiting time - How much time processes spend in the ready queue waiting for their turn to get on the CPU.

- **Load average** - The average weighting time of processes in the ready queue waiting for their turn to get into the CPU. It is reported in 1-minute, 5-minute, and 15-minute averages by "uptime" and "who".

Response time - The time taken in an interactive program from the issuance of a command to the commence of a response to that command [5].

Context switches - This allows multiple processes to share a single CPU which is of the essence in a multitasking operating system. It stores the state of a process or a thread to be restored and resume the execution from the same point later [6].

To improve the performance of the round-robin CPU scheduling algorithm by adding significant changes. In the past few years, there have been different papers that include various features to overcome the limitations of the classical round-robin algorithm. Hahne used a MinMax dispersion measure by the remaining CPU burst time to calculate the time quantum for the processes [7]. A team of researchers proposed a method utilizing the median and variance to carry out a round-robin scheduling algorithm with a dynamic time quantum [8]. Another uses a comparative approach by sorting the burst time of the jobs in ascending order and then executing it [9].

These different studies have suggested different emergence to beat the shortcomings of the conventional round-robin scheduling algorithm that is presently being used. Foremost the Round Robin (RR) CPU Scheduling Algorithm will have a comparison against the different means to compute for the quantum time. In this paper quantum time in each cycle is calculated based on the different means that is useful when burst times of processes are varies and some of them have a difference from the other. For all sets that contain at least one pair of unequal value, the harmonic mean is the lowest value of them, whereas the arithmetic mean is always highest of them and the geometric mean is always between them. Suppose that we have n processes with burst time t1, t2, tn. Arithmetic mean, geometric mean, and harmonic mean [10] is shown in the following equations below:

$$Arithmetic\ mean\ =\ \frac{t_1 + t_2 + ... + t_n}{n}$$
$$Geometric\ mean\ =\ \sqrt{t_1\ +\ t_2\ +...\ +\ t_n}$$
$$Harmonic\ mean\ =\ \frac{n}{\frac{1}{t_1} + \frac{1}{t_2} + ... + \frac{1}{t_n}}$$

The researchers present the burst times of processes varies using the different mean to have an impact on reducing the average turnaround time, average waiting time, response time and context switches.

METHODOLOGY

I. Pseudocode of the Round Robin Algorithm using Different Means

In our algorithm, the arrival time of all processes is assumed to be zero. The inputs of the algorithm are burst time and number of processes (n). qt is the quantum time.

1. All the present processes are assigned to ready queue.
2. While(ready queue is not empty)
3. Calculate quantum time using the different means:

$$q_t = Arithmetic\ Mean\ of\ Burst\ Times$$

or

$$q_t = Geometric\ \ Mean\ of\ Burst\ Times$$

or

$$q_t = Harmonic\ Mean\ of\ Burst\ Times$$

4. Assign qt to processes

$$P_i \leftarrow q_t$$
$$i\ =\ i + 1$$

5. If (i<number of processes) then goto step 4
6. If a new process is arrived:

   Update ready queue and goto step 3

7. Calculate average turnaround time, average waiting time, response time and context switches
8. End

Ii. Illustration of the Round Robin Algorithm using Different Means

To further illustrate how the algorithm works, we present a simulation. Table 1 considers five processes. The burst times of these processes are given. The arrival time of these processes is considered zero. Burst times have been considered so that the burst time of the fifth process had a large difference with the previous processes.

TABLE 1. Sample Processes

Process	Arrival Time	Burst Time
P1	0	40
P2	0	60
P3	0	20
P4	0	55
P5	0	5

Here, we show the steps of calculating quantum time for Table 1 by using different means. Take notes that the numbers have been rounded.

Arithmetic Mean

$$q_t \ in \ step \ 1 = \frac{40+60+20+55+5}{5} = 36$$
$$q_t \ in \ step \ 2 = \frac{4+24+0+19+0}{3} = 15.66 \approx 16$$
$$q_t \ in \ step \ 3 = \frac{0+8+0+0+3}{2} = 5.5 \approx 6$$
$$q_t \ in \ step \ 4 = \frac{0+2+0+0+0}{1} = 2$$

Geometric Mean

$$q_t \ in \ step \ 1 = \sqrt{40+60+20+55+5} = 13.42 \approx 13$$
$$q_t \ in \ step \ 2 = \sqrt{27+47+7+42+0} = 11.09 \approx 11$$
$$q_t \ in \ step \ 3 = \sqrt{16+36+0+31+0} = 9.11 \approx 9$$
$$q_t \ in \ step \ 4 = \sqrt{7+27+0+22+0} = 7.48 \approx 7$$
$$q_t \ in \ step \ 5 = \sqrt{0+20+0+15+0} = 5.91 \approx 6$$
$$q_t \ in \ step \ 6 = \sqrt{0+12+0+9+0} = 4.58 \approx 5$$
$$q_t \ in \ step \ 7 = \sqrt{0+7+0+4+0} = 3.31 \approx 3$$
$$q_t \ in \ step \ 8 = \sqrt{0+4+0+1+0} = 2.24 \approx 2$$
$$q_t \ in \ step \ 9 = \sqrt{0+2+0+0+0} = 1.41 \approx 1$$
$$q_t \ in \ step \ 10 = \sqrt{0+1+0+0+0} = 1$$

Harmonic Mean

$$q_t \, in \, step \; 1 = \frac{5}{\frac{1}{40} + \frac{1}{60} + \frac{1}{20} + \frac{1}{55} + \frac{1}{5}} = 16.13 \approx 16$$

$$q_t \, in \, step \; 2 = \frac{4}{\frac{1}{24} + \frac{1}{44} + \frac{1}{4} + \frac{1}{39} + 0} = 11.76 \approx 12$$

$$q_t \, in \, step \; 3 = \frac{3}{\frac{1}{12} + \frac{1}{32} + 0 + \frac{1}{27} + 0} = 19.58 \approx 20$$

$$q_t \, in \, step \; 4 = \frac{2}{0 + \frac{1}{12} + 0 + \frac{1}{7} + 0} = 8.84 \approx 9$$

$$q_t \, in \, step \; 5 = \frac{1}{0 + \frac{1}{3} + 0 + 0 + 0} = 3$$

REFERENCES

[1] Singh, A, Goyal P, and Batra S. 2010. *An Optimized Round Robin Scheduling Algorithm for CPU Scheduling*. International Journal on Computer Science and Engineering (IJCSE). vol **2**. pp **2383-85**.

[2] Silberchatz A, Galvin P B, and Gagne G. 2003. *Operating systems concepts.*

https://pdfs.semanticscholar.org/780d/07fbf09484a2490868d45255a23762a7e21d.pdf

[3] Al-Hagery M. 2011. *A selective Quantum of Time for Round Robin Algorithm to Increase CPU Utilization.* vol **3**. pp **54-9**.

https://www.researchgate.net/publication/327894111_A_selective_Quantum_of_Time_for_Round_Robin_Algorithm_to_Increase_CPU_Utilization

[4] Ayeni B, Afolayan O, and Yusuf N. 2017. *Improved Dynamic Time Slice Round Robin Scheduling Algorithm with Unknown Burst Time*. International Journal of Computer Applications. vol **167** no **7**. pp 21-2.

https://pdfs.semanticscholar.org/780d/07fbf09484a2490868d45255a23762a7e21d.pdf

[5] Silberschatz A, Gagne G, and Galvin PB. 2009. *Operating System Concepts (8th. ed.)* John Wiley & Sons Incorporation, United States, US.

https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_CPU_Scheduling.html

[6] Silberschatz A,Gagne G, and Galvin PB. 2013. *Operating Systems Concept (Courier Kendallville).* pp **261-6.**

https://www.researchgate.net/publication/331658405_Fittest_Job_First_D

ynamic_Round_Robin_FJFDRR_scheduling_algorithm_using_dual_queu
e_and_arrival_time_factor_a_comparison

[7] Hahne E. 2012. *An effective round robin algorithm using Min-Max dispersion measure.* International Journal on Computer Science and Engineering (IJCSE). vol **4** no **1**

[8] Behera HS, Mishra M, and Patel S. 2012. *Weighted clustering based preemptive scheduling for real time system*. International Journal on Computer Science and Engineering (IJCSE). vol **4** no **5**

[9] Behera HS, Mohanty R, Sahu S, and Bhoi SK. 2011. Comparative performance analysis of multi-dynamic time quantum round robin (mdtqrr). Indian Journal of Computer Science and Engineering. vol **2** no **2**.
https://pdfs.semanticscholar.org/5780/7683807ec0bddb703d886b
4d68c017678405.pdf?_ga=2.26494333.1122857007.1569902619-25227
5181.1569902619

[10] Jamal A and Zubair A. 2012. *A Varied Round Robin Approach using Harmonic Mean of the Remaining Burst Time of the Processes.*
https://www.researchgate.net/publication/307816684_A_New_Met
hod_to_Improve_Round_Robin_Scheduling_Algorithm_with_Qua
ntum_Time_Based_on_Harmonic-Arithmetic_Mean_HARM