Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023
DOI : https://doi.org/10.33395/sinkron.v8i4.12051
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Average Max Round Robin Algorithm: A Case Study

Tri Dharma Putra[1], Rakhmat Purnomo[2]*
[1,2]Department of Informatics, Faculty of Computer Science, Universitas Bhayangkara Jakarta Raya
tri.dharma.putra@dsn.ubharajaya.ac.id, rakhmat.purnomo@dsn.ubharajaya.ac.id

Submitted : Dec 22, 2022 | **Accepted** : Jan 22, 2023 | **Published** : Oct 1, 2023

Abstract: Round Robin Algorithm is one wellknown algorithm in real time system. Several variants of round robin algorithms are in the fields. Average max round robin algorithm is a breakthrough to optimize context switching or also called quantum. Context switching is one big problem in round robin algorithm. To optimize high context switching is the key solution. This will make this algorithm efficient. There should be a way to optimize this context switching. Then the average max round robin algorithm is one solution to this problem. The average max algorithm is defined by finding the average of burst time then add the maximum burst time to the average burst time. Then calculate again the average of the two. The calculation will be iterated in the next round robin cycle. Here, in this journal, three case studies are discussed. Each with different burst times to understand this average max round robin algorithm more clearly. In the first case study we get turn around time 34 ms, and average waiting time 20.6 ms. In the second case study, we get average turn around time 21.8 ms, and the average waiting time 13 ms. And in the last case study, the third one, we get turn around time 12.2 ms, and the average waiting time 6.6 ms. There is no calculation for the second iteration for all case studies. Since the left burst time is only in one process. Optimizing the context switching, minimizing average turnaround time, and average waiting time is the key solution to round robin algorithm.

Keywords: Average Max Round Robin Algorithm, Average Turn Around Time, Average Waiting Time, Context Switching, Quantum

INTRODUCTION

In operating systems, multitasking and multiprocessing are two terms that are used frequently. If more CPUs are used for processing, then it is called multiprocessing (Purnomo & Putra, 2022) (Purnomo & Putra, 2022) (Sakshi et al., 2022). On the contrary, if one CPU is used for processing, and that programs switch between each other so fast, it is known as multitasking. However in this case, the user feels that programs are running simultaneously.

Scheduling is a major task in operating system (Omar et al., 2021). The methods of scheduling play an important role in terms of system's performance (Tri Dharma Putra, 2021). There are several scheduling algorithm in operating system. Round robin, FCFS (First Come First Serve), Priority, Shortest Job First (SJF), and its variants (Harki et al., 2020).

In terms of multitasking, operating systems scheduling is a critical duty. A ready queue is maintained if more than one process need to be executed (Putra, 2022) (Tri Dharma Putra, 2021) (Abu-Dalbouh, 2022). If the computer has two processors, each processor has its own ready queue. The very idea of efficiency is to keep waiting time, turn around time as minimum as possible. This is the main goal of research interests, namely, to minimize time cost (Putra & Purnomo, 2021) (Putra & Purnomo, 2022) (Mostafa & Amano, 2020).

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i4.12051 p-ISSN : 2541-044X

There are several algorithms in CPU scheduling, and usually implemented in different methods. The most common scheduling algorithm in real time systems is preemptive round robin. Round robin scheduling gives each process time quantum, which is also called context switching (Putra, 2022). After its process time expires, process will exit, and another process will be executed. This iteration for process will be done iteratively. Until all processes is finished (Ali et al., 2021).

In average max round robin algorithm, the time slice or time quantum is dynamic. After every iteration of process's execution, the time quantum or sometimes called context switching is calculated again. So that in every iteration, there is a great possibility that the context switching will be different. That is the dynamic time quantum. The idea of making the context switching dynamic in every iteration cycle is a breakthrough in this average max round robin algorithm. Since every cycle or every iteration has different burst time, making dynamic context switching is important. Average max round robin algorithm is usually abbreviated as AMRR algorithm.

LITERATURE REVIEW

Uferah Shafi and friends presented a novel amended dynamic round robin scheduling algorithm for timeshared systems. Which is also called ADRR. The main goal of this round robin algorithm is to improved conventional round robin algorithm by utilizing active quantum time notion. Quantum time was cyclically adjusted based on CPU burst time (Shafi et al., 2020). By using MATLAB simulation results it was revealed that ADRR outperformed other wellknown round robin algorithm such as Priority Based Round Robin (PRR), Improved Round Robin (IRR), conventional Round Robin, and Optimum Multilevel Dynamic Round Robin (OMDRR).

Alaa Fiad and friends proposed an improved round robin scheduling algorithm based on analytic model. The use of burst times as parameter in their model ensure a more suitable context switching. This algorithm proved can be applied in many operating systems, but also can be applied in cloud computing environment (Fiad et al., 2020). It was showed that this approach improved the turn around time and waiting time.

Fahd Alhaidari and Balharith proposed, an enhanced round robin algorithm in cloud computing environment for optimal process scheduling. The two proposed a novel technique called dynamic round robin heuristic algorithm (DRRHA) by tuning the context switching in a dynamic ways based on the mean of context switching (Alhaidari & Balharith, 2021). This research uses CloudSim Plus tool that proved that DRRHA outperformed significantly other round robin algorithm in terms of response time, turnaround time, and waiting time.

Daniela and Vitor proposed, a systematic mapping of round robin scheduling algorithm. This is a new developments in round robin applications. This journal identifies the state of the art in terms of round robin researches (Freire et al., 2021). This research will guide practitioners and researchers in the field of operating systems and CPU scheduling.

Mostafa proposed a modified version of RR algorithm, called dynamic time slide (DTS), to combine the advantageous of the overhead which is low scheduling of RR and find better short process for the sake of minimizing time cost. Each process has a weight proportional to the weights of all processes. This weight in process, determines its time slice within current period. Each process in the cluster is assigned the average of processes time slice in this cluster (Mostafa & Amano, 2020).

Shahad conducted a research review in CPU scheduling, by comparing several wellknown algorithms and finding the best algorithm. The idea of this comparative study is sugest various ways to improve CPU optimization criteria through different algorithms to improve waiting time, response time, turnaround time (Ali et al., 2021). The conclusion was there was no algorithm better in all criteria.

METHODS

Here, the structure of this journal is comprised of six sections. The first section is introduction. In this introduction, the basic idea and the background concept of scheduling process algorithm are given. Also the research gap of this AMRR is given. The second section is literature review. Previous works by other experts are explained here. The third section is research method. In this section, the concept of method is presented thoroughly, especially the main concept of average max round robin algorithm. The

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023
DOI : https://doi.org/10.33395/sinkron.v8i4.12051
e-ISSN : 2541-2019
p-ISSN : 2541-044X

fourth section is result. Here, three case studies are given and discussed. The fifth section is discussion. Comparison result of the three case studies are given here. The the last chapter is conclusion and future works.

Let discuss the average max round robin algorithm by example. Let say we have four processes. The arrival times of these four are all zero. And the CPU burst times of these four processes are given as A1=2, A2 = 8, A3=4, and A4 =6. Please take a look at Table 1. below:

Table 1. Process and Burst Time

Process	Burst Time
A1	2
A2	8
A3	4
A4	6

Then we set them in ascending order and calculate it to find the context switching. The formula is :

Context Switching = (Average + Maximum Burst Time)/2

So the average burst time = (2 + 8 + 4 + 6)/4= 5. Then we calculate the maximum burst time with this average and afterwards calculate the average between di two. Then the context switching = (5+8)/2 = 6,5. This is rounded up, so that the context switching is 7. The process will be running in ascending order (Sakshi et al., 2022). The first process, A1 is executed first, this is 2 ms. Then A3 will be executed, after that A4 and finally A2 is executed.

After each iteration completes, context switching is calculated again by sorting the process in ascending order. This iteration will continue until all processes are finished.

RESULTS

Below is three case studies to analyse this average max round robin algorithm. Each case study consists of five processes. Gantt chart is presented to understand this algorithm more clearly. Also table to calculate the average turn around time and average waiting time.

First Case Study

In this first case study, let's say we have five processes namely P1 to P5. The processes must be sorted to be ascending. Let's say arrival time all is zero. Please take a look at Table 2. below:

Table 2. Process and Burst Time

Process	Burst Time
P1	8
P2	10
P3	12
P4	17
P5	20

To calculate the quantum, we must find the average of these five processes. The average is (8+10+12+17+20)/5, which is equal to 13.4. Then we find the largest burst time which is P5, with burst time 20. We add this average with the largest burst time, which is (13.4 + 20)/2, it equals to 16.7. Then we get the quantum to be rounded up above, 17.

The gantt chart of this process is presented as below, in Figure 1.:

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023
DOI : https://doi.org/10.33395/sinkron.v8i4.12051
e-ISSN : 2541-2019
p-ISSN : 2541-044X

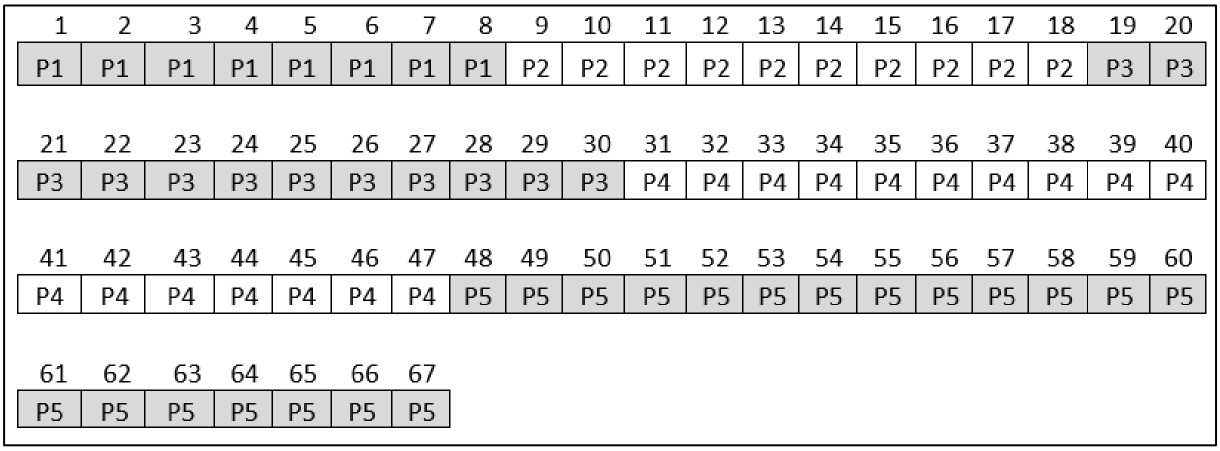

Fig. 1 Gantt Chart of first case study

In the first iteration, P1 will be finished, since the quantum is 17 and the burst time of P1 is only 8 ms. Then at 9 ms, P2 gets in. It will be processed for 10 ms until 18 ms. At 19 ms, P3 started to be executed until 30 ms, since the burst time of P3 is 12 ms. Then at 31 ms, P4 is executed until 47 ms. Since the burst time of P4 is 17 ms. Then finally P5 is executed until 64 ms. In the second iteration, P1, P2, P3, and P3 are already done. Since in the ready queue only left P5, which is 3 ms. Then P5 continues to be executed until 67 ms. There is no calculation for the second iteration since, the process only left for P5, only one process.

To calculate turn around time, please take a look at the Table 3. below. The table must be populated first.

Table 3. Calculation of Turnaround Time

Process	Arrival Time	Burst time	Start Time	Finish Time	Turn Around Time
P1	0	8	0	8	8
P2	0	10	8	18	18
P3	0	12	18	30	30
P4	0	17	30	47	47
P5	0	20	47	67	67

So that, it is concluded that we get average turn around time: 34 ms. To calculate waiting time is as follows:

P1 = 0
P2 = 8
P3 = 18
P4 = 30
P5 = 47

So that the average waiting time: 20,6

Second Case Study

In this second case study, let's say we have five processes namely P1 to P5. The processes must be sorted to be ascending. Let's say the arrival time all is zero. Please take a look at Table 4. below:

Table 4. Process and Burst Time

Process	Burst Time

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023 e-ISSN : 2541-2019
DOI : https://doi.org/10.33395/sinkron.v8i4.12051 p-ISSN : 2541-044X

P1	4
P2	6
P3	10
P4	11
P5	13

To calculate the quantum, we must find the average of these five processes. The average is (4+6+10+11+13)/5, which is equal to 8.8. Then we find the largest burst time which is P5, with burst time 13. We add this average with the largest burst time, which is (8.8 + 13)/2, it equals to 10.9. Then we get the quantum to be rounded up above, to be 11.

The gantt chart of these processes are presented as below, in Figure 2.:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1	P1	P1	P1	P2	P2	P2	P2	P2	P2	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
P4	P4	P4	P4	P4	P4	P4	P4	P4	P4	P4	P5	P5	P5	P5	P5	P5	P5	P5	P5
41	42	43	44																
P5	P5	P5	P5																

Fig. 2 Gantt Chart of Second Case Study

In the first iteration, P1 will be finished, since the quantum is 11 and the burst time of P1 is only 4 ms. Then at 5 ms, P2 gets in. It will be processed for 6 ms until 10. At 11 ms, P3 started to be executed until 20 ms, since the burst time of P3 is 10 ms. Then at 21 ms, P4 is executed until 31 ms. Since the burst time of P4 is 11 ms. Then finally P5 is executed until 42 ms. In the second iteration, P1, P2, P3, and P3 are already done. Since in the ready queue only left P5, which is 2 ms. Then P5 continues to be executed from 42 until 44 ms. Since the process only left for one process, namely P5, then there is no calculation for second iteration cycle.

To calculate turn around time, please take a look at the Table 5. below. The table must be populated first.

Table 5. Calculation of Turnaround Time

Process	Arrival Time	Burst time	Start Time	Finish Time	Turn Around Time
P1	0	4	0	4	4
P2	0	6	4	10	10
P3	0	10	10	20	20
P4	0	11	20	31	31
P5	0	13	31	44	44

So that, it is concluded that we get average turn around time: 21.8 ms. To calculate waiting time is as follows:

P1 = 0
P2 = 4
P3 = 10
P4 = 20
P5 = 31

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023
DOI : https://doi.org/10.33395/sinkron.v8i4.12051
e-ISSN : 2541-2019
p-ISSN : 2541-044X

So that the average waiting time: 13 ms.

Third Case Study

In this third case study, let's say we have five processes namely Q1 to Q5. The processes must be sorted to be ascending. Let's say the arrival time all is zero. Please take a look at Table 6. below:

Table 6. Process and Burst Time

Process	Burst Time
Q1	2
Q2	3
Q3	5
Q4	6
Q5	7

To calculate the quantum, we must find the average of these five processes. The average is (2+3+5+6+7)/5, which is equal to 4.6. Then we find the largest burst time which is Q5, with burst time 7. We add this average with the largest burst time, which is (4.6 + 7)/2, it equals to 5.8. Then we get the quantum to be rounded up above, to be 6.
The gantt chart of these processes are presented as below, in Figure 3.:

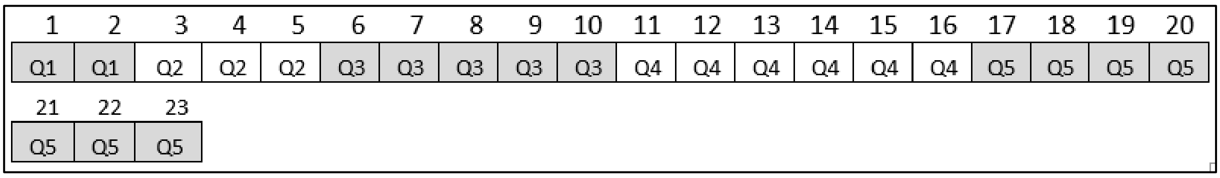

Fig. 3 Gantt Chart of Third Case Study

In the first iteration, Q1 will be finished, since the quantum is 6 and the burst time of Q1 is only 2 ms. Then at 3 ms, Q2 gets in. It will be processed for 3 ms until 5 ms. At 6 ms, Q3 started to be executed until 10 ms, since the burst time of Q3 is 5 ms. Then at 11 ms, Q4 is executed until 16 ms. Since the burst time of Q4 is 6 ms. Then finally Q5 is executed until 22 ms. In the second iteration, Q1, Q2, Q3, and Q4 are already done. Since in the ready queue only left Q5, which is 1 ms. Then Q5 continues to be executed from 22 until 23 ms. There is no calculation for second iteration cycle, since the process only left one process, namely P5, which is 1 ms.

To calculate turn around time, please take a look at the Table 7. below. The table must be populated first.

Table 7. Calculation of Turnaround Time

Process	Arrival Time	Burst time	Start Time	Finish Time	Turn Around Time
Q1	0	2	0	2	2
Q2	0	3	2	5	5
Q3	0	5	5	10	10
Q4	0	6	10	16	16
Q5	0	7	16	23	23

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Number 4, October 2023
DOI : https://doi.org/10.33395/sinkron.v8i4.12051
e-ISSN : 2541-2019
p-ISSN : 2541-044X

So that, it is concluded that we get average turn around time: 12.2 ms. To calculate waiting time is as follows:
Q1 = 0
Q2 = 2
Q3 = 5
Q4 = 10
Q5 = 16
So that the average waiting time: 6,6 ms.

DISCUSSION

In the first, second and the third case study, in the second iteration, there is only one processs left, so that there is no calculation for the second iteration. It is concluded that there is no second calculation of burst times with average max calculations. Context switching for the first case study is 17, second case study is 11, and third case study is 6. Total burst time is 67 for first case study, 44 for second case study and 23 for third case study. Average turn around time is 34 ms for first case study, 21.8 ms for second case study and 23 ms for the third case study. Average waiting time is 20.6 ms for the first case study, 13 ms for the second case study, and 6.6 ms for the third case study.

The table comparison between the three case studies is below, in Table 8.:

Table 8. Comparison between Case Studies

	Calculation for Second Iteration	Context Switching (first iteration)	Total Burst Time	Average Turn Round Time	Average Waiting Time
First Case Study	No	17	67	34	20.6
Second Case Study	No	11	44	21.8	13
Third Case Study	No	6	23	12.2	6.6

CONCLUSION

In these three case studies, we do not find the second calculation of context switching in the second iteration. Since, the process in each case study, only left one process. In average max round robin which is also called AMRR, the iteration is important, since this will change and optimize the context switching after each iteration. If we compare this AMRR algorithm with conventional round robin, in term of context switching, then we can see that the context switching is reduced. In the first case study we get turn around time 34 ms, and average waiting time 20.6 ms. In the second case study we get turn around time 21.8 ms, and the average waiting time 13 ms. In the third case study we get average turn around time 12.2 ms and average waiting time 6.6 ms. The context switching for the first case study is 17. The context switching for second case study is 11, and for the third case study is 6. For future works, may be comparison with other round robin algorithms can be done and analyze.

REFERENCES

Abu-Dalbouh, H. M. (2022). A New Combination Approach to CPU Scheduling based on Priority and Round-Robin Algorithms for Assigning a Priority to a Process and Eliminating Starvation. *International Journal of Advanced Computer Science and Applications*, *13*(4), 541–546. https://doi.org/10.14569/IJACSA.2022.0130463

Alhaidari, F., & Balharith, T. Z. (2021). Enhanced round-robin algorithm in the cloud computing environment for optimal task scheduling. *Computers*, *10*(5).

*name of corresponding author

https://doi.org/10.3390/computers10050063

Ali, S. M., Alshahrani, R. F., Hadadi, A. H., Alghamdi, T. A., Almuhsin, F. H., & El-Sharawy, E. E. (2021). A Review on the CPU Scheduling Algorithms: Comparative Study. *International Journal of Computer Science & Network Security*, *21*(1), 19–26. https://doi.org/10.22937/IJCSNS.2021.21.1.4

Fiad, A., Maaza, Z. M., & Bendoukha, H. (2020). Improved version of round robin scheduling algorithm based on analytic model. *International Journal of Networked and Distributed Computing*, *8*(4), 195–202. https://doi.org/10.2991/IJNDC.K.200804.001

Freire, D. L., Frantz, R. Z., Roos-frantz, F., & Fernandes, V. B. (2021). *New developments in Round Robin algorithms and their applications : a systematic mapping study New developments in Round Robin algorithms and their applications : a systematic mapping study Daniela L . Freire , Rafael Z . Frantz , Fabricia Roos-Frantz Vit. January*.

Harki, N., Ahmed, A., & Haji, L. (2020). CPU Scheduling Techniques: A Review on Novel Approaches Strategy and Performance Assessment. *Journal of Applied Science and Technology Trends*, *1*(2), 48–55. https://doi.org/10.38094/jastt1215

Mostafa, S. M., & Amano, H. (2020). An adjustable variant of round robin algorithm based on clustering technique. *Computers, Materials and Continua*, *66*(3), 3253–3270. https://doi.org/10.32604/cmc.2021.014675

Omar, H. K., Jihad, K. H., & Hussein, S. F. (2021). Comparative analysis of the essential cpu scheduling algorithms. *Bulletin of Electrical Engineering and Informatics*, *10*(5), 2742–2750. https://doi.org/10.11591/eei.v10i5.2812

Purnomo, R., & Putra, T. D. (2022). *Simulation of Preemptive Shortest Job First Algorithm*. *11*(5), 1–11. https://doi.org/10.17148/IJARCCE.2022.11501

Putra, T. D. (2022). Analysis of Priority Preemptive Scheduling Algorithm: Case Study. *Ijarcce*, *11*(1), 27–30. https://doi.org/10.17148/ijarcce.2022.11105

Putra, T. D., & Purnomo, R. (2021). Analisis Algoritma Round Robin pada Penjadwalan CPU. *Jurnal Ilmiah Teknologi Informasi Asia*, *15*(2), 85. https://doi.org/10.32815/jitika.v15i2.481

Putra, T. D., & Purnomo, R. (2022). Case Study : Improved Round Robin Algorithm. *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, *7*(3), 950–956.

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

Shafi, U., Shah, M., Wahid, A., Abbasi, K., Javaid, Q., Asghar, M., & Haider, M. (2020). A novel amended dynamic round robin scheduling algorithm for timeshared systems. *International Arab Journal of Information Technology*, *17*(1), 90–98. https://doi.org/10.34028/iajit/17/1/11

Tri Dharma Putra, A. F. (2021). Comparison Between Simple Round Robin and Intelligent Round Robin Algorithms in CPU Scheduling. *International Journal of Advanced Research in Computer and Communication Engineering*, *10*(4), 86–90.