Round Robin Algorithm With Average Quantum Dynamic Time Based on Multicore Processor

Emma Rosita Simarmata1,3, Gortap Lumbantoruan1,4, Rena Nainggolan1, Junika Napitupulu2

1Department of Komputerisasi Akuntansi, Universitas Methodist Indonesia, Medan, Indonesia
2Department of Manajemen Informatika, Universitas Methodist Indonesia, Medan, Indonesia
3emros2016@gmail.com
4lumbantoruan.gortap@gmail.com

Abstract Modern operating systems that are widely used today, have used a multiprogramming system and use a multicore system to solve multiprogramming. Multicore systems provide greater efficiency than a single processor running on a single chip. The efficiency of the multicore system in completing its tasks is very dependent on the CPU scheduling algorithm used by the operating system. Round Robin algorithm is one of the CPU scheduling that is widely used in multiprogramming. With this algorithm, the CPU will divide the time for each process given the same time until the entire process in the queue is complete. In this study, the researcher improvised the Round Robin algorithm on a multicore processor system to execute the processes contained in a single queue queue that is to give a quantum time based on the average bursttime of the entire process in the queue. In order to minimize waiting time and turnaround time throughout the process.

1. Introduction
Modern operating systems, have used a multiprogramming system and use a multicore system to solve multiprogramming. The efficiency of the multicore system in completing its tasks is very dependent on the CPU scheduling algorithm used by the operating system. Various CPU scheduling algorithms include Round Robin, FCFS (First Come, First Serve), SJF (Shortest Job First), Priority Scheduling[4]

Journal of A High Performance Real-Time Multicore Load Balance Strategy for applying Mixed-Earliest Deadline First (MEDF) to overcome the problem of sending loads and saving power in real-time systems and perfecting Enhanced Deadline-Driven Dynamic Voltage and Frequency Scaling (ED3VFS) parameters for increase power and improve processor performance. And the result is the proposed algorithm can reduce energy consumption up to 54.2% and the performance of the proposed algorithm is superior to the others [5].

Other researchers also examined the Round Robin algorithm used in CPU scheduling with the aim of obtaining waiting times from smaller processes such as research entitled "Modification of Round Robin Algorithms with Quantum Time Dynamic and Ordering Process Ascending" analyzing the use of dynamic quantum time on scheduling Round Robin algorithm. Dynamic quantum time is obtained through the bursttime average of the process in the queue. And the result is that waiting time and
turnaround time are smaller than the classic Round Robin algorithm[6]. Noon et al applies the dynamic quantum time method on Round Robin scheduling using the mean average, the results obtained are obtained by waiting time and context switching which is smaller than the classic round robin method[7].

The study, entitled "Efficient and Scalable Multiprocessor Fair Scheduling Using Distributed Weighted Round-Robin" presents a solution to the problem of fairness in CPU scheduling. DWRR is based on distributed thread queues and small additional overhead for schedulers. DWRR allows users to specify weights to threads and achieve an accurate proportional CPU division with a constant error limit[8].

Kumar et al did an optimization approach in order to minimize the context switch in the Round Robin algorithm, and concluded that the number of switching contexts for a series of processes could be reduced by getting a new quantum size using the simplex algorithm. And the result is the number of context switching will decrease along with the number of increases / decreases in waiting time and context switching of each process. So as to minimize overhead costs [9].

Another research to improve CPU performance with the Round Robin algorithm is to integrate with other scheduling algorithms. Combining classic scheduling algorithms for multicore processor systems is combining Shortest Job First with Round Robin and concluding that the average waiting time value is smaller than the classic Round Robin in each case of uniprocessor and multi-processor[10].

2. Multicore System

Processing with parallelism systems has been used since the beginning of computing was created with the aim of improving performance. Multicore systems are computers with two or more different physical processors, and they are able to run real parallel programs. Here, with additional hardware costs, performance gains can be achieved by executing parallel processes in different processors[11].

3. Multiprocessor Scheduling

Scheduling on the Multiprocessor that is being investigated is a multiprocessor, a multiprogramming environment that includes various systems. Some special characteristics of multiprogramming environments, which influence the type of scheduling strategies are the type of processor and processor affinity[14].

![Figure 1. Parallel Execution in a Multiprocessor System [12]](image-url)

4. Round Robin Algorithm

Round Robin algorithm is designed for time sharing systems. This algorithm is similar to FCFS scheduling, but preemption is added to switch between processes. The queue ready and allocates each process for a certain time interval until a quantum slice time [15]

5. Research Methodology

In this study, we will discuss how to build scheduling for multicore processor systems using a single queue system. The scheduling process used is Round Robin algorithm where the given quantum time is dynamically.
5.1 Data Used
In this study, the authors need some input data consisting of:
1. Quantity of Core Processor
2. Quantity of Process
3. Burst Time

5.2 The Round Robin Algorithm Procedure with Dynamic Quantum Time
The modified Round CPU scheduling algorithm is given dynamic quantum time where quantum time is obtained from the calculation of the average of the total bursttime of the entire process that is in a single queue.

\[
\text{Quantum Time} = \frac{P_1 + P_2 + \ldots + P_n}{n}
\]

5.3 Procedure Placement Process to Core
Round Robin takes the first process of the queue and allocates the CPU to a time interval of up to 1 time quantum. After completing the quantum processing time, the control unit checks the remaining CPU time from the current process. If the remaining CPU time is left from the process that is running less than 1 time quantum, the CPU is re-allocated to the current process for the remaining CPU time remaining. In this case the process will complete the execution and will be deleted from the queue. The scheduler then proceeds to the next process in the queue until the entire process is completed.
5.4 Assessment Parameters
The throughput which is the parameter that is assessed is the Average Waiting Time (AWT) and Average Turnaround Time (ATT). Average Waiting Time is the average waiting time of all processes waiting in line until the entire process has been executed. The formula used to get the Average Waiting Time is

$$AWT = \frac{\text{Waiting Time} (P_1+P_2+\ldots+P_n)}{n}$$ (2)

Where as to look for Average Turnaround Time is

$$ATT = \text{Waiting Time} (P_1 + P_2 + \ldots + P_n) + \text{Total Burst Time}$$ (3)

6. Results and Discussion
In this study, the experiment was carried out on a multicore processor system that had 4 (four) cores and all cores were independent. The process that is queued will be entered into a single queue system, all processes have the same priority. Context Switching equals zero means that there is no context switching that occurs when switching from one process to another. All processes are tied to the CPU. There is no I/O bound process. Quantum time is taken in milliseconds.

6.1 Process Queue Test Results with 5 Processes
In this trial the arrival time of the process has been considered zero. With the number of processes contained in the queue that uses the single queue system there are five processes and will be served 4 (four) core processors.

| Table 1. Process Queue with 5 Processes |
|----------------------------------------|
| Proses      | BT  |
| P1          | 12  |
| P2          | 5   |
| P3          | 8   |
| P4          | 10  |
| P5          | 13  |

6.2 Trial Results 5 Process with Improve Round Robin Algorithm
Test by modifying the Round Robin algorithm, which is done first is determined quantum time by looking for the average burst time value of the entire process.

1) Determine Quantum Time

Quantum time = \((BT \ P_1 + BT \ P_5 + BT \ P_3 + BT \ P_4 + BT \ P_2) / 5\)

= \((12 + 5 + 8 + 10 + 13) / 5\)

= \(48 / 5\)

= \(9.5 \rightarrow 10\)

2) Mapping of Core Processor Assignments

| Table 2. Mapping the assignment of Core Processors with Improve Round Robin |
|-----------------------------|-----|-----|-----|
| Core 0         | R1  | R2  | R3  |
| P1            | P1  | -   | P3  |
| Core 1        | P2  | P5  | -   |
| Core 2        | P3  | P1  | -   |
| Core 3        | P4  | -   |     |
3) **Average Turnaround Time (ATT)**

| Proses | Burst time | Waiting time | Turnaround Time | Average Turnaround Time |
|--------|------------|--------------|-----------------|-------------------------|
| P1     | 12         | 0            | 12              |                         |
| P2     | 5          | 0            | 5               |                         |
| P3     | 8          | 0            | 8               | 10.6                    |
| P4     | 10         | 0            | 10              |                         |
| P5     | 13         | 5            | 18              |                         |

4) **Classic Round Robin Comparison Chart and Improve Round Robin for 5 Processes**

Figure 4. Waiting Time Comparison Graph

For comparison of turnaround time using the classic Round Robin algorithm and Improve Round Robin both almost have similar graphs. This is because waiting time with the two algorithms is the same, which is a total of 5 milliseconds.

Figure 5. Graph of Turnaround Time Comparison

6.3 **Analysis Results**

After conducting the experiments described above, the authors compare average waiting time and average turnaround time using the Round Robin Classic and Improve Round Robin. In the Round Robin Classic algorithm quantum time is a random number determined by the operating system starting from numbers 1-100, while Improve Round Robin searches for quantum time based on the average bursttime of the entire process.

| Jumlah Proses | Round Robin Klasik | RR Modifikasi | Persentase Penurunan |
|---------------|--------------------|---------------|----------------------|
|               | QT | AWT | ATT | QT | AWT | ATT | AWT (%) | ATT (%) |
| 5             | 4  | 1   | 10,6| 10 | 1   | 10,6| 0       | 0       |
| 10            | 4  | 10  | 22  | 12 | 7,2 | 19,2| 28      | 12,72   |
| 15            | 4  | 17,4667 | 27,4667| 10 | 16,2 | 26,2| 7,25    | 4,61    |
Based on the table above, it can be concluded that Improve Round Robin algorithm uses quantum time based on average bursttime, has average waiting time and average waiting time is smaller than Round Robin which uses static quantum time.

7. Conclusion
Based on the discussion that has been done in the previous chapter, the researcher can conclude that using Improve Round Robin algorithm produces a smaller waiting time so that it will reduce CPU overhead when compared to using the classic Round Robin algorithm and Improve Round Robin algorithm can produce context switching less

References
[1] Lidya, M. and Nicolas, Benedict Jayaprakash. An Efficient Partitioned Sheduling Mechanism for Real Time Task Execution in a Multicore System. International Journal of Latest Trends in Engineering and Technology (IJLTET), 7, 3 (Juny 2016), pp. 215-222.
[2] Priyadarsani, S., Rajalakshmi, V., and Muneeswari, Dr.G. AFT Scheduling Algorithm for Multicore Architecture: A Novel Approach. International Research Journal of Engineering and Technology (IRJET), 03, 03 (March 2016). pp. 1243-1247.
[3] A. Vajda, “Multi-core and Many-core Processor Architectures,” in Programming Many-Core Chips, Boston, MA: Springer US, 2011, pp. 9–43.
[4] P. Jaipurkar and P. D. Tembhurne, “A Comparative Study of Different Customized Multiprocessor Scheduling Algorithms on Multicore Architecture,” vol.5, no.3. p. 5.
[5] J. C. Saez, M. Prieto, A. Fedorova, dan S. Blagodurov, “A comprehensive scheduler for asymmetric multicore systems,” in Proceedings of the 5th European conference on Computer systems - EuroSys ’10, Paris, France, 2010, p. 139.
[6] G. Lumbantoruan, “Modifikasi Algoritma Round Robin Dengan Dynamic Quantum Time Dan Pengurutan Proses Secara Ascending,” p. 12, 2016.
[7] A. Noon, A. Kalakech, and S. Kadry, “A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average,” vol. 8, no. 3. p. 6, 2011.
[8] T. Li, D. Baumberger, and S. Hahn, “Efficient and Scalable Multiprocessor Fair Scheduling Using Distributed Weighted Round-Robin,” p. 10.
[9] Manonmaniam Sundararman University, A. R. Kumar, and C. C. sekar, “A Dynamic Prioritized Load Balanced Round Robin Elasticity Mechanism on Cloud Mobile Services,” Int. J. Comput. Trends Technol., vol. 32, no. 1. pp. 54–61, Jan 2016.
[10] M. P. Shau, “Scheduling in Multi-core Systems: Minimizing Average Waiting Time by merging (Round-Robin with Shortest-Job-First Technique),” vol. 5, no. 3. p. 3, 2017.
[11] S. Plana and F. Xhafa, Ed., Programming multicore and many-core computing systems, First edition. Hoboken, New Jersey: Wiley, 2017.
[12] Silberschatz, A., Galvin, P.B., and Gagne, G. Operating System Concepts. John Wiley & Sons, New Jersey, 2012.
[13] W. Zhang, “Design and Implementation of Multi-core Support for an Embedded Real-time Operating System for Space Applications,” p. 87.
[14] M. Naghibzadeh, Operating system concepts and techniques. New York: iUniverse, 2005.
[15] Stallings, W. Operating System: Internal and Design Principles. Prentice Hall, New Jersey, 2012.
[16] A. Gea, “Optimasi Turn Around Time Pada Penjadwalan Round Robin Dengan Mencari Quantum Time Optimal Menggunakan Algoritma Simulated Annealing,” METHODIKA: Jurnal Teknik Informatika dan Sistem Informasi, vol. 1, no. 1, 2017.