2022-2023 Spring

CS 342 – Operating Systems

Project #2 — Report

Berk Çakar - 22003021 - Section 2

Kutay Tire - 22001787 - Section 3

Data Tables:

For each possible scheduling approach, algorithm, and the number of processors (we skipped 2, 4, 6, 8 and used 1, 3, 5, 7, 9) we ran the mps program three times and took the average of the three corresponding average turnaround times. Instead of an input file, we used the random generation with the same parameters in order to prevent the bias that can be caused by the previously determined input set (i.e., a particular set of inputs might perform better for an algorithm or scheduling approach).

We run the following commands:

```
./mps -n 1 -a S NA -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 3 -a S NA -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 5 -a S NA -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 7 -a S NA -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 9 -a S NA -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 1 -a M RM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 3 -a M RM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 5 -a M RM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 7 -a M RM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 9 -a M RM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 1 -a M LM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 3 -a M LM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 5 -a M LM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 7 -a M LM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 9 -a M LM -s FCFS 0 -m 1 -r 30 10 50 300 200 400 50
```

```
./mps -n 1 -a S NA -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 3 -a S NA -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 5 -a S NA -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 7 -a S NA -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 9 -a S NA -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 1 -a M RM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 3 -a M RM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 5 -a M RM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 7 -a M RM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 9 -a M RM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 1 -a M LM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 3 -a M LM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 5 -a M LM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 7 -a M LM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
./mps -n 9 -a M LM -s SJF 0 -m 1 -r 30 10 50 300 200 400 50
```

```
./mps -n 1 -a S NA -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 3 -a S NA -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 5 -a S NA -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 7 -a S NA -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 9 -a S NA -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 1 -a M RM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 3 -a M RM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 5 -a M RM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 7 -a M RM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 9 -a M RM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 1 -a M LM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 3 -a M LM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 5 -a M LM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 7 -a M LM -s RR 20 -m 1 -r 30
10 50 300 200 400 50

./mps -n 9 -a M LM -s RR 20 -m 1 -r 30
10 50 300 200 400 50
```

```
./mps -n 1 -a S NA -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 3 -a S NA -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 5 -a S NA -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 7 -a S NA -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 9 -a S NA -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 1 -a M RM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 3 -a M RM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 5 -a M RM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 7 -a M RM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 9 -a M RM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 1 -a M LM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 3 -a M LM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 5 -a M LM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 7 -a M LM -s RR 40 -m 1 -r 30
10 50 300 200 400 50

./mps -n 9 -a M LM -s RR 40 -m 1 -r 30
10 50 300 200 400 50
```

Resulting output is the following:

No. of Processors	Scheduling Algorithm and Approach											
	FCFS			SJF			RR					
	S	RM	LM	S	RM	LM	Q = 20 (S | RM | LM)			Q = 40 (S | RM | LM)		
1	6798.50	6647.74	6592.50	5997.48	5743.44	5754.84	13313.18	13226.64	13143.43	12235.50	12150.72	12515.66
3	1993.04	1961.54	1856.98	1740.46	1699.52	1633.66	3602.66	3527.28	3276.88	3206.32	3138.26	2956.26
5	1067.52	1045.16	987.08	947.96	939.76	932.58	1868.34	1781.16	1751.86	1809.56	1737.82	1677.74
7	683.88	663.48	640.72	633.28	626.34	593.82	1135.04	1043.22	1040.04	1036.06	1017.98	991.20
9	482.08	449.68	491.16	487.08	481.00	470.38	663.64	583.66	672.40	610.58	565.92	681.80

Table 1: Average turnaround time outputs for different scenarios. Values are in milliseconds.

Plots and Explanations:

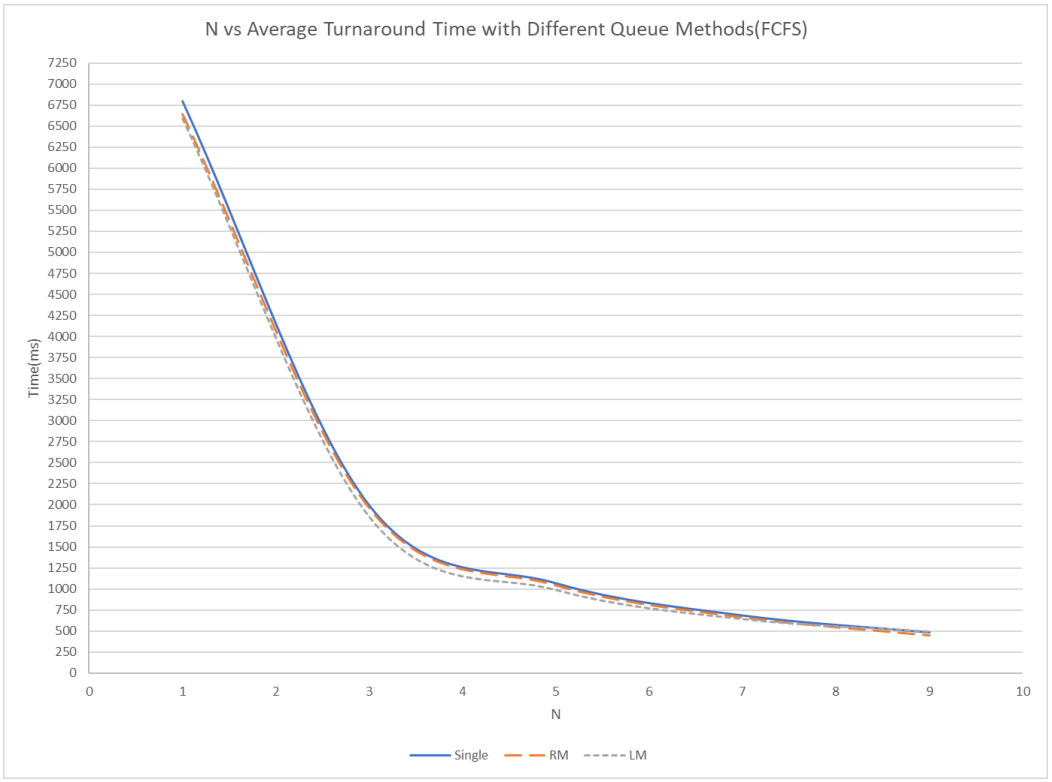

Figure 1

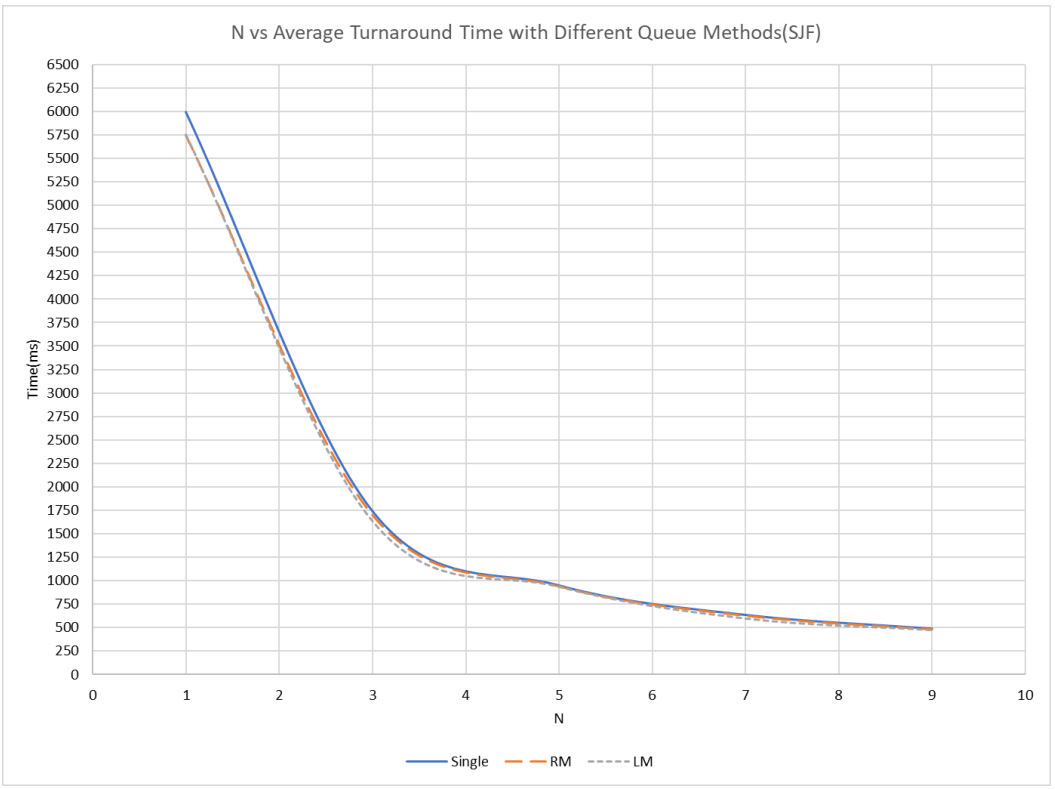

Figure 2

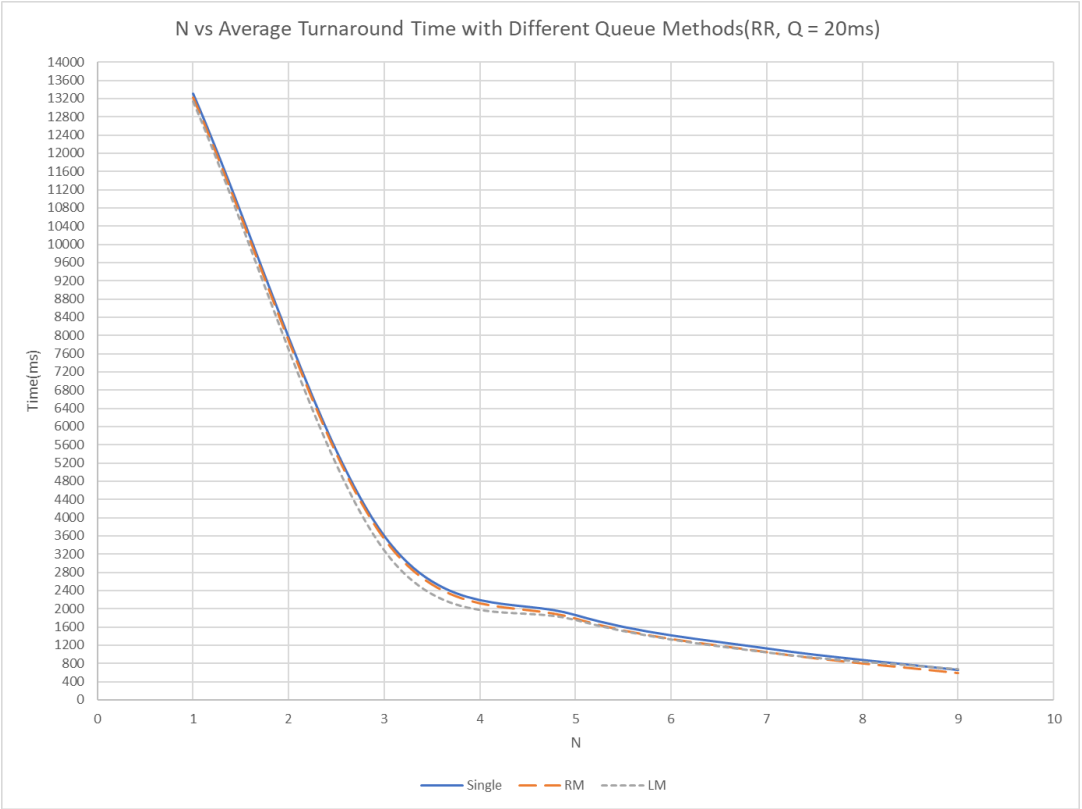

Figure 3

In terms of the single-queue approach vs. the multi-queue approach, a significant improvement has been observed in average turnaround times with both LM and RM methods. As figures 1,2 and 3 demonstrate, the average turnaround times of LM and RM methods are less than the single-queue approach in every scheduling algorithm. This is not surprising as, in the single-queue approach, the process goes to whatever processor is available without any proper selection method. On the other hand, when the multi-queue approach is used, the processes are distributed more evenly. Naturally, these differences were better observed as the number of processes increased. Hence, 50 processes were used to obtain the data and compare the different approaches. Furthermore, as the number of process threads (N) increased, the average turnaround time decreased in all approaches. This is expected because there are simply more processors to process and especially the respective queues of each processor become less loaded in the multi-queue approach. Overall, with enough processes, the multi-queue approach definitely was more efficient.

In terms of RM and LM methods, LM functioned slightly better. Observing this difference was not easy as there was no clear way of controlling a thread switch caused by the computer. The results were mostly similar, but again when a large number of processes were used, LM was more advantageous. This may be due to the reason that RM distributes the processes based on their process ids without checking the load of each queue whereas LM distributed them by selecting the minimum loaded one during insertion to the queue.

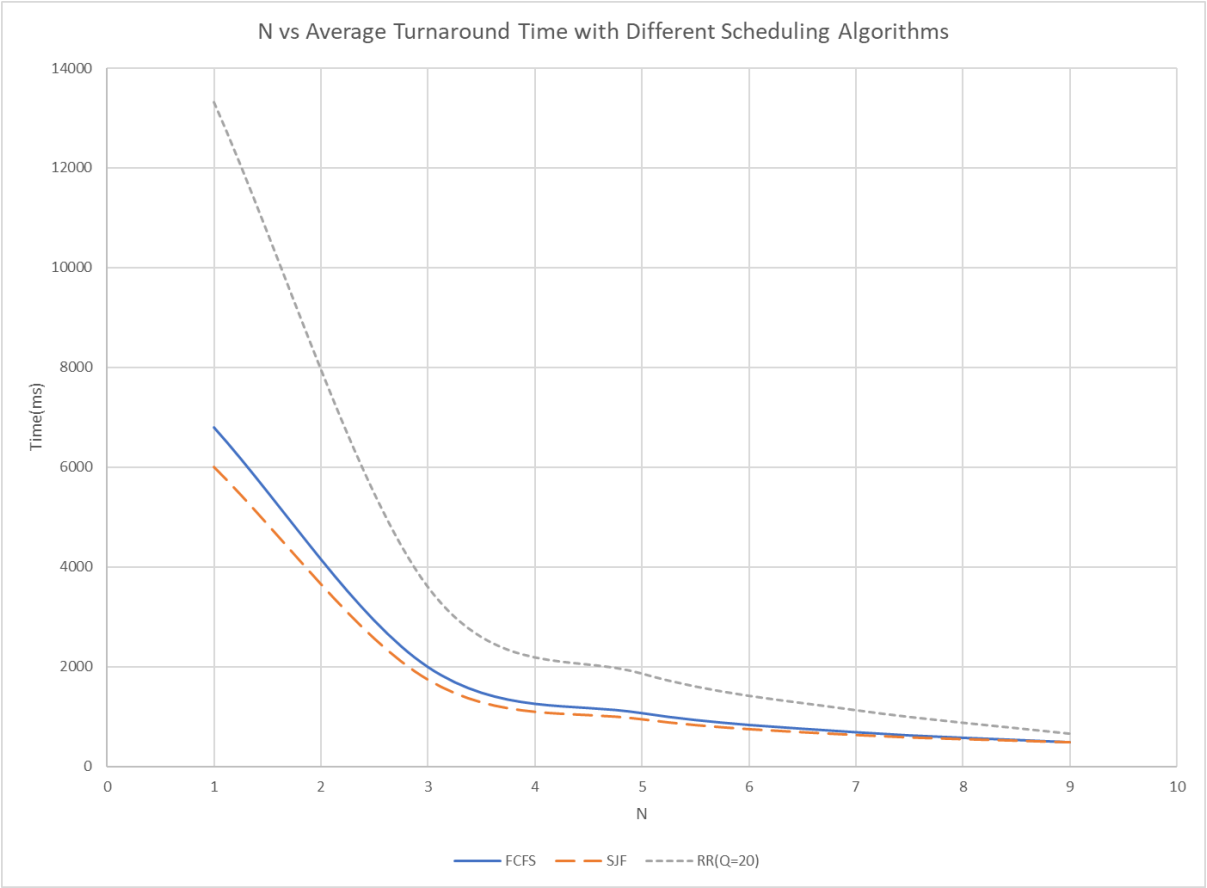

Figure 4

All three algorithms were compared in the single queue approach. As expected, when the number N is small, SJF was the best-performing one in terms of average turnaround time and waiting time. As the process with the shorter burst length is scheduled first, it is expected SJF to have a lower average waiting and turnaround time. On the other hand, RR gave the largest average turnaround time. This is because when the time slice of a process expires, it is sent to the back of the queue without allowing the process to finish. Therefore, on average, the processes finish later than when SJF or FCFS is used.

In general, SJF also had a smaller average turnaround time than FCFS. The difference between FCFS and SJF in terms of average turnaround time could be more especially if the processes with larger burst lengths arrive first in FCFS. This is a well-known problem called the convoy effect, but as we used randomized IATs and burst lengths, the effect was not apparent. Nevertheless, SJF still managed to outperform FCFS even though the difference seemed to be smaller with higher numbers of N as figure 4 also suggests. Similarly, in both of the three approaches, the average turnaround time decreased. As explained earlier, this is attributed to the fact that there are simply more available processor threads to process when N is larger.

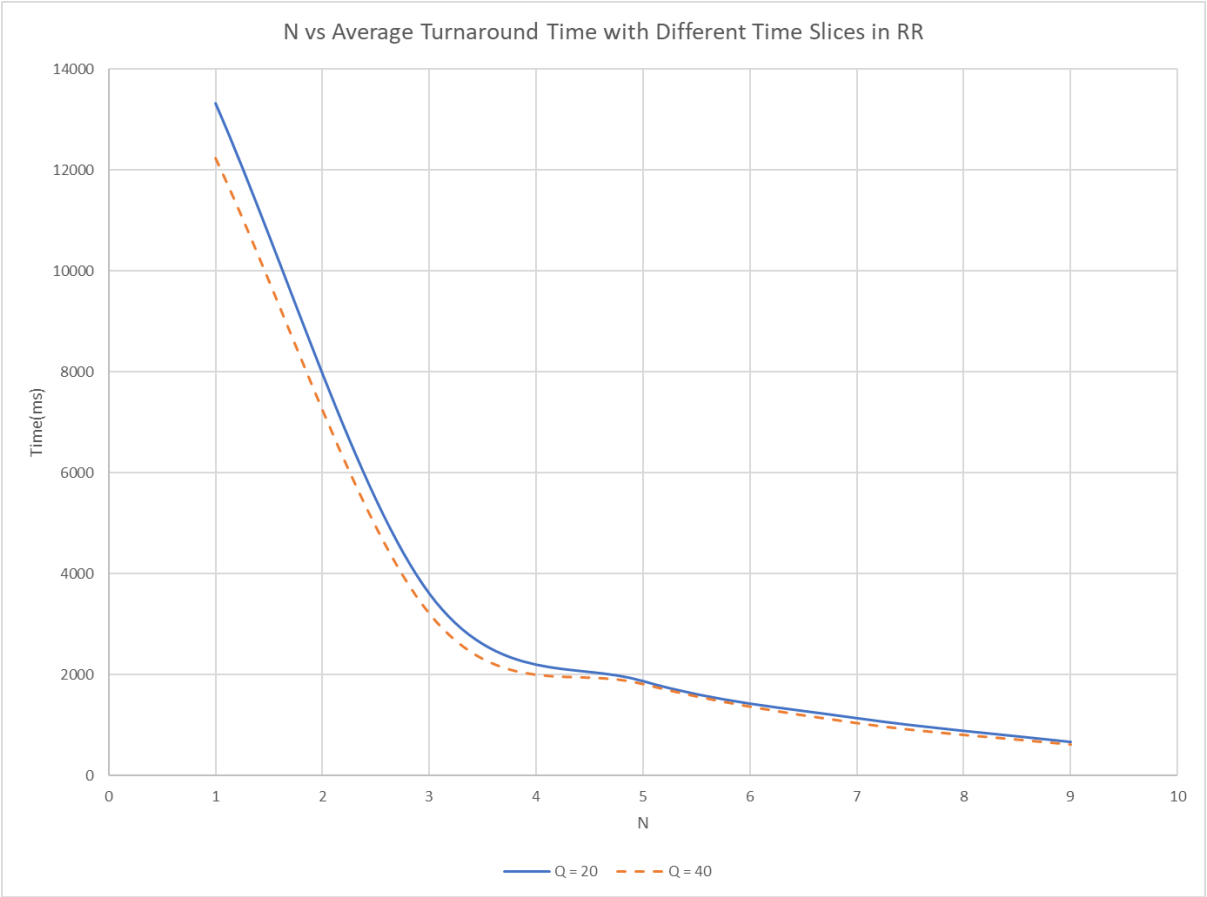

Figure 5

Finally, using longer time-quantum led to a smaller average turnaround time for round-robin scheduling. As it can also be seen from figure 5, when q is 20, the average turnaround time is higher than the case when q is 40. This is normal because when the time quantum is too small, there will be a lot of context switches which could cause a longer turnaround time.