
Simulation of a Task Scheduler (M1 CSA - Report)

Author(s): Florian Legendre



Contents

I	Monoserver Scheduling	2
1	FIFO	3
1.1	Implementation	3
1.2	Scheduling results	3
1.3	Metrics	4
2	Round Robin	5
2.1	Implementation	5
2.2	Scheduling results	5
2.3	Metrics	7
3	EDF	8
3.1	Implementation	8
3.2	Scheduling results	8
3.3	Metrics	9
4	RMS	10
4.1	Implementation	10
4.2	Scheduling results	10
4.3	Metrics	11
5	Comparison Table	12
II	Multiserver Scheduling	13
6	EDF	14
6.1	Implementation	14
6.2	Scheduling results	14
6.3	Metrics	15
7	RMS	16
7.1	Implementation	16
7.2	Scheduling results	16
7.3	Metrics	17
8	Comparison Table	18
III	Multiserver Energy-aware Scheduling	19
9	FIFO	20
9.1	Implementation	20
9.2	Scheduling results	20
9.3	Metrics	21
10	EDF	22
10.1	Implementation	22
10.2	Scheduling results	22
10.3	Metrics	23
11	Comparison Table	24

Part I

Monoserver Scheduling

FIFO

1.1 Implementation

1.2 Scheduling results

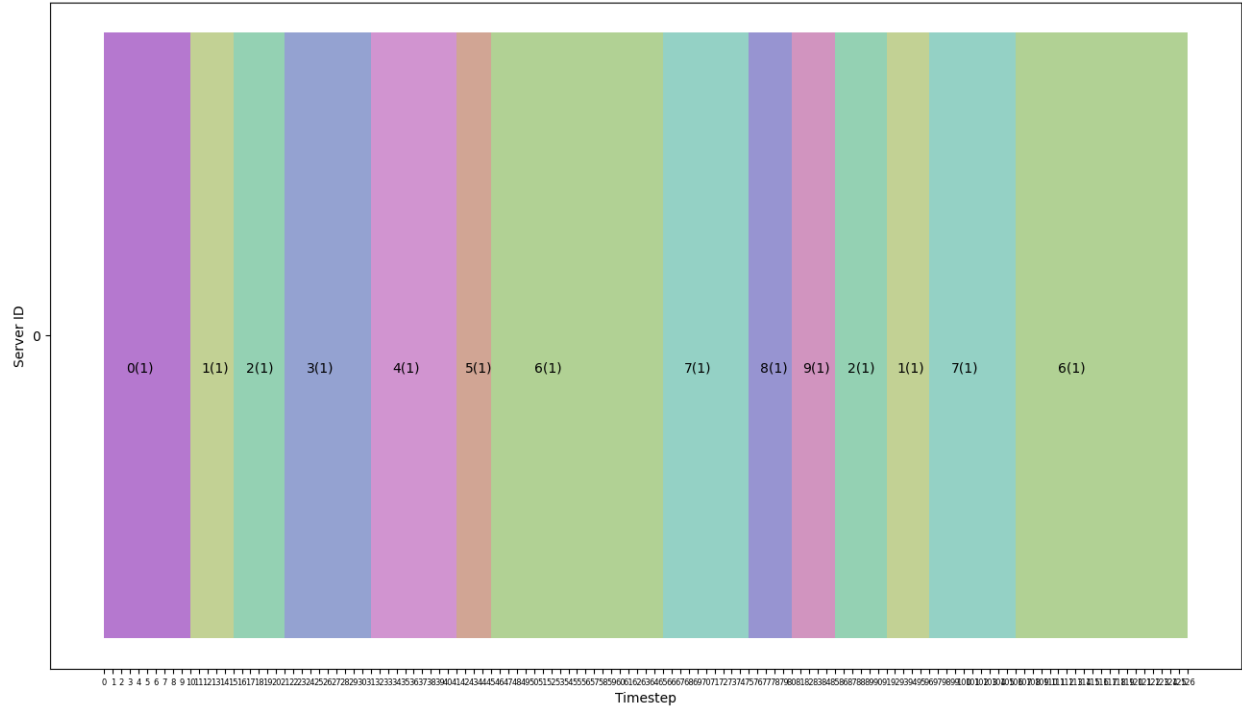


Figure 1: Output schedule produced by the FIFO scheduling algorithm on one server

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

0	0	0.0	10.0	1.0
1	0	10.0	15.0	1.0
2	0	15.0	21.0	1.0
3	0	21.0	31.0	1.0
4	0	31.0	41.0	1.0
5	0	41.0	45.0	1.0
6	0	45.0	65.0	1.0
7	0	65.0	75.0	1.0
8	0	75.0	80.0	1.0
9	0	80.0	85.0	1.0
2	0	85.0	91.0	1.0
1	0	91.0	96.0	1.0
7	0	96.0	106.0	1.0
6	0	106.0	126.0	1.0

Listing 1: Detail of the scheduling result

1.3 Metrics

```
##### SCHEDULE METRICS #####

>> Scheduling Metrics:
- Total Makespan: 126.0
- Nb Deadline Misses: 11
- Max Tardiness: 69.0
- Average Tardiness: 36.63636363636363
- Late Jobs:
[Job: J6{6.0a/0.0u/30.0rd/36.0ad/100.0p} | Tardiness: 29.0 ]
[Job: J2{3.0a/0.0u/10.0rd/13.0ad/10.0p} | Tardiness: 8.0 ]
[Job: J2{13.0a/0.0u/10.0rd/23.0ad/10.0p} | Tardiness: 68.0 ]
[Job: J8{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 64.0 ]
[Job: J4{5.0a/0.0u/30.0rd/35.0ad/0.0p} | Tardiness: 6.0 ]
[Job: J7{10.0a/0.0u/25.0rd/35.0ad/50.0p} | Tardiness: 40.0 ]
[Job: J7{60.0a/0.0u/25.0rd/85.0ad/50.0p} | Tardiness: 21.0 ]
[Job: J1{0.0a/0.0u/10.0rd/10.0ad/20.0p} | Tardiness: 5.0 ]
[Job: J5{6.0a/0.0u/12.0rd/18.0ad/0.0p} | Tardiness: 27.0 ]
[Job: J9{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 69.0 ]
[Job: J1{20.0a/0.0u/10.0rd/30.0ad/20.0p} | Tardiness: 66.0 ]

>> Servers Metrics:
- Servers work load:
Server #0 : 126.0

>> Energy Metrics:
- Total Consumption: 2799.999999999995
- Max Consumption: 22.22222222222222
- Average Consumption: 2799.999999999995
- Consumption per Server:
Server #0 : 2799.999999999995

#####
```

Listing 2: Metrics for FIFO on a single server

Round Robin

2.1 Implementation

2.2 Scheduling results

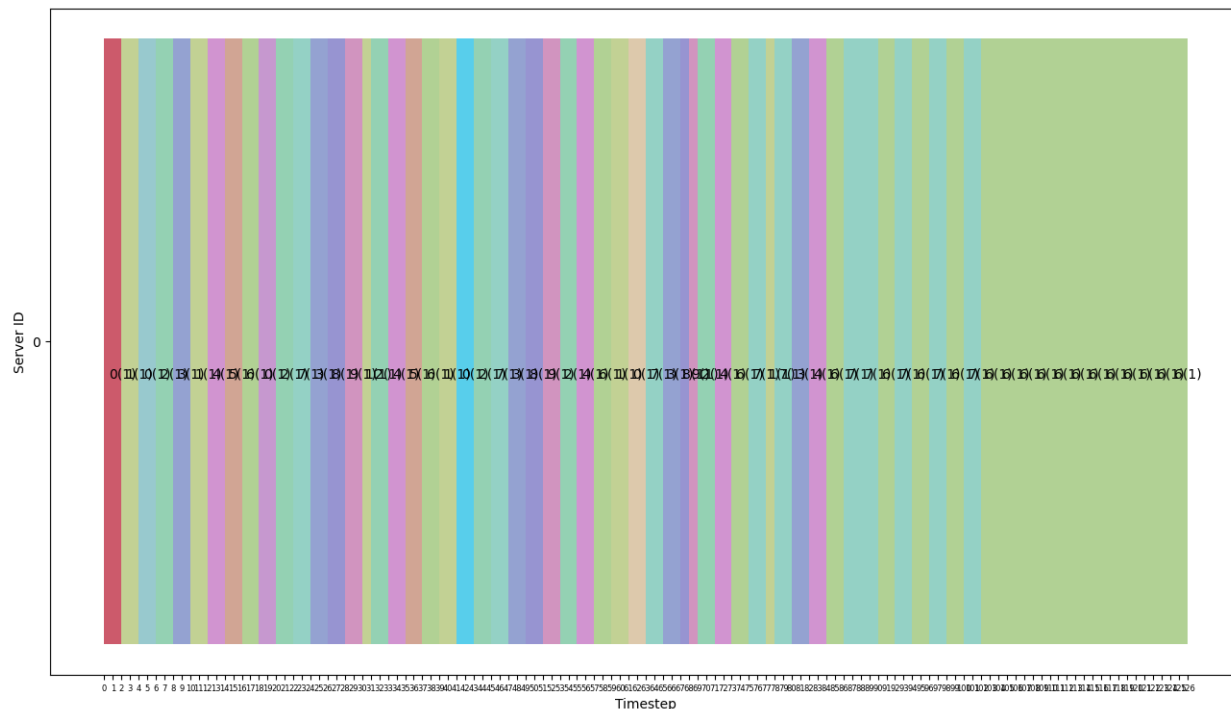


Figure 2: Output schedule produced by the Round Robin scheduling algorithm on one server

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

0	0	0.0	2.0	1.0
1	0	2.0	4.0	1.0
0	0	4.0	6.0	1.0
2	0	6.0	8.0	1.0
3	0	8.0	10.0	1.0
1	0	10.0	12.0	1.0
4	0	12.0	14.0	1.0
5	0	14.0	16.0	1.0
6	0	16.0	18.0	1.0
0	0	18.0	20.0	1.0
2	0	20.0	22.0	1.0
7	0	22.0	24.0	1.0
3	0	24.0	26.0	1.0
8	0	26.0	28.0	1.0
9	0	28.0	30.0	1.0
1	0	30.0	31.0	1.0
2	0	31.0	33.0	1.0
4	0	33.0	35.0	1.0
5	0	35.0	37.0	1.0
6	0	37.0	39.0	1.0
1	0	39.0	41.0	1.0
0	0	41.0	43.0	1.0
2	0	43.0	45.0	1.0

7	0	45.0	47.0	1.0
3	0	47.0	49.0	1.0
8	0	49.0	51.0	1.0
9	0	51.0	53.0	1.0
2	0	53.0	55.0	1.0
4	0	55.0	57.0	1.0
6	0	57.0	59.0	1.0
1	0	59.0	61.0	1.0
0	0	61.0	63.0	1.0
7	0	63.0	65.0	1.0
3	0	65.0	67.0	1.0
8	0	67.0	68.0	1.0
9	0	68.0	69.0	1.0
2	0	69.0	71.0	1.0
4	0	71.0	73.0	1.0
6	0	73.0	75.0	1.0
7	0	75.0	77.0	1.0
1	0	77.0	78.0	1.0
7	0	78.0	80.0	1.0
3	0	80.0	82.0	1.0
4	0	82.0	84.0	1.0
6	0	84.0	86.0	1.0
7	0	86.0	88.0	1.0
7	0	88.0	90.0	1.0
6	0	90.0	92.0	1.0
7	0	92.0	94.0	1.0
6	0	94.0	96.0	1.0
7	0	96.0	98.0	1.0
6	0	98.0	100.0	1.0
7	0	100.0	102.0	1.0
6	0	102.0	104.0	1.0
6	0	104.0	106.0	1.0
6	0	106.0	108.0	1.0
6	0	108.0	110.0	1.0
6	0	110.0	112.0	1.0
6	0	112.0	114.0	1.0
6	0	114.0	116.0	1.0
6	0	116.0	118.0	1.0
6	0	118.0	120.0	1.0
6	0	120.0	122.0	1.0
6	0	122.0	124.0	1.0
6	0	124.0	126.0	1.0

Listing 3: Detail of the scheduling result

2.3 Metrics

```
##### SCHEDULE METRICS #####  
  
>> Scheduling Metrics:  
- Total Makespan: 126.0  
- Nb Deadline Misses: 13  
- Max Tardiness: 70.0  
- Average Tardiness: 43.07692307692308  
- Late Jobs:  
[Job: J0{0.0a/0.0u/15.0rd/15.0ad/0.0p} | Tardiness: 48.0 ]  
[Job: J9{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 53.0 ]  
[Job: J7{10.0a/0.0u/25.0rd/35.0ad/50.0p} | Tardiness: 55.0 ]  
[Job: J5{6.0a/0.0u/12.0rd/18.0ad/0.0p} | Tardiness: 19.0 ]  
[Job: J8{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 52.0 ]  
[Job: J1{0.0a/0.0u/10.0rd/10.0ad/20.0p} | Tardiness: 21.0 ]  
[Job: J3{4.0a/0.0u/30.0rd/34.0ad/0.0p} | Tardiness: 48.0 ]  
[Job: J4{5.0a/0.0u/30.0rd/35.0ad/0.0p} | Tardiness: 49.0 ]  
[Job: J2{13.0a/0.0u/10.0rd/23.0ad/10.0p} | Tardiness: 48.0 ]  
[Job: J1{20.0a/0.0u/10.0rd/30.0ad/20.0p} | Tardiness: 48.0 ]  
[Job: J2{3.0a/0.0u/10.0rd/13.0ad/10.0p} | Tardiness: 32.0 ]  
[Job: J6{6.0a/0.0u/30.0rd/36.0ad/100.0p} | Tardiness: 70.0 ]  
[Job: J7{60.0a/0.0u/25.0rd/85.0ad/50.0p} | Tardiness: 17.0 ]  
  
>> Servers Metrics:  
- Servers work load:  
Server #0 : 126.0  
  
>> Energy Metrics:  
- Total Consumption: 2799.999999999995  
- Max Consumption: 22.22222222222222  
- Average Consumption: 2799.999999999995  
- Consumption per Server:  
Server #0 : 2799.999999999995  
  
#####
```

Listing 4: Metrics for Round Robin on a single server

EDF

3.1 Implementation

3.2 Scheduling results

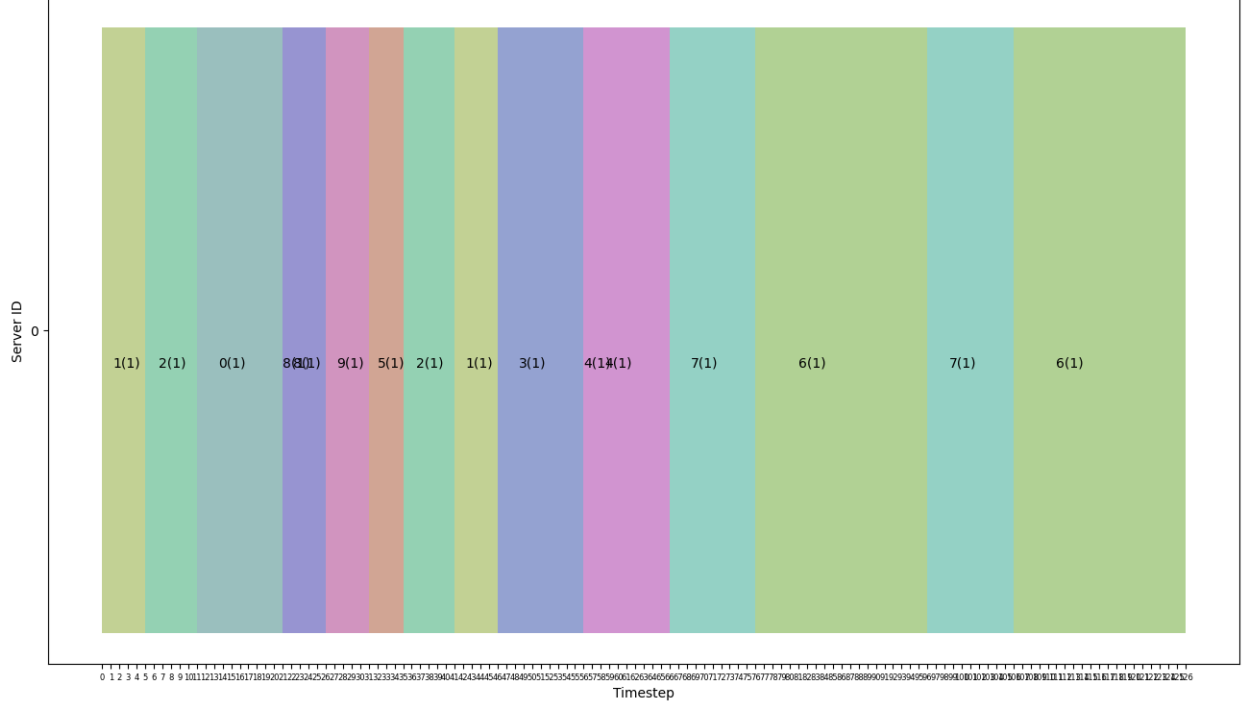


Figure 3: Output schedule produced by the EDF scheduling algorithm on one server

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

```
1 0 0.0 5.0 1.0
2 0 5.0 11.0 1.0
0 0 11.0 21.0 1.0
8 0 21.0 21.0 1.0
8 0 21.0 26.0 1.0
9 0 26.0 31.0 1.0
5 0 31.0 35.0 1.0
2 0 35.0 41.0 1.0
1 0 41.0 46.0 1.0
3 0 46.0 56.0 1.0
4 0 56.0 56.0 1.0
4 0 56.0 66.0 1.0
7 0 66.0 76.0 1.0
6 0 76.0 96.0 1.0
7 0 96.0 106.0 1.0
6 0 106.0 126.0 1.0
```

Listing 5: Detail of the scheduling result

3.3 Metrics

```
##### SCHEDULE METRICS #####

>> Scheduling Metrics:
- Total Makespan: 126.0
- Nb Deadline Misses: 11
- Max Tardiness: 60.0
- Average Tardiness: 23.363636363636363
- Late Jobs:
[Job: J2{13.0a/0.0u/10.0rd/23.0ad/10.0p} | Tardiness: 18.0 ]
[Job: J6{6.0a/0.0u/30.0rd/36.0ad/100.0p} | Tardiness: 60.0 ]
[Job: J3{4.0a/0.0u/30.0rd/34.0ad/0.0p} | Tardiness: 22.0 ]
[Job: J5{6.0a/0.0u/12.0rd/18.0ad/0.0p} | Tardiness: 17.0 ]
[Job: J0{0.0a/0.0u/15.0rd/15.0ad/0.0p} | Tardiness: 6.0 ]
[Job: J9{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 15.0 ]
[Job: J7{60.0a/0.0u/25.0rd/85.0ad/50.0p} | Tardiness: 21.0 ]
[Job: J4{5.0a/0.0u/30.0rd/35.0ad/0.0p} | Tardiness: 31.0 ]
[Job: J8{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 10.0 ]
[Job: J7{10.0a/0.0u/25.0rd/35.0ad/50.0p} | Tardiness: 41.0 ]
[Job: J1{20.0a/0.0u/10.0rd/30.0ad/20.0p} | Tardiness: 16.0 ]

>> Servers Metrics:
- Servers work load:
Server #0 : 126.0

>> Energy Metrics:
- Total Consumption: 2799.999999999995
- Max Consumption: 22.22222222222222
- Average Consumption: 2799.999999999995
- Consumption per Server:
Server #0 : 2799.999999999995

#####
```

Listing 6: Metrics for EDF on a single server

RMS

4.1 Implementation

4.2 Scheduling results

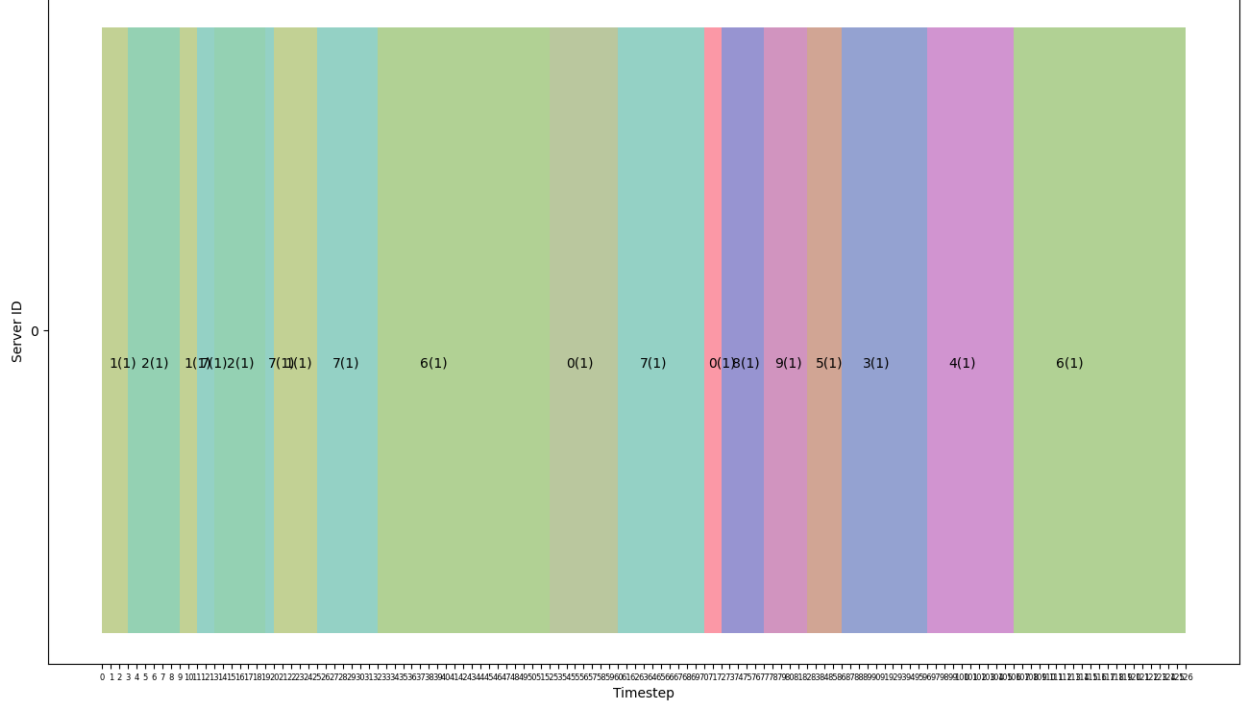


Figure 4: Output schedule produced by the RMS scheduling algorithm on one server

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

1	0	0.0	3.0	1.0
2	0	3.0	9.0	1.0
1	0	9.0	11.0	1.0
7	0	11.0	13.0	1.0
2	0	13.0	19.0	1.0
7	0	19.0	20.0	1.0
1	0	20.0	25.0	1.0
7	0	25.0	32.0	1.0
6	0	32.0	52.0	1.0
0	0	52.0	60.0	1.0
7	0	60.0	70.0	1.0
0	0	70.0	72.0	1.0
8	0	72.0	77.0	1.0
9	0	77.0	82.0	1.0
5	0	82.0	86.0	1.0
3	0	86.0	96.0	1.0
4	0	96.0	106.0	1.0
6	0	106.0	126.0	1.0

Listing 7: Detail of the scheduling result

4.3 Metrics

```
##### SCHEDULE METRICS #####
>> Scheduling Metrics:
- Total Makespan: 126.0
- Nb Deadline Misses: 8
- Max Tardiness: 71.0
- Average Tardiness: 50.25
- Late Jobs:
[Job: J4{5.0a/0.0u/30.0rd/35.0ad/0.0p} | Tardiness: 71.0 ]
[Job: J1{0.0a/0.0u/10.0rd/10.0ad/20.0p} | Tardiness: 1.0 ]
[Job: J6{6.0a/0.0u/30.0rd/36.0ad/100.0p} | Tardiness: 16.0 ]
[Job: J8{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 61.0 ]
[Job: J5{6.0a/0.0u/12.0rd/18.0ad/0.0p} | Tardiness: 68.0 ]
[Job: J9{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 66.0 ]
[Job: J0{0.0a/0.0u/15.0rd/15.0ad/0.0p} | Tardiness: 57.0 ]
[Job: J3{4.0a/0.0u/30.0rd/34.0ad/0.0p} | Tardiness: 62.0 ]

>> Servers Metrics:
- Servers work load:
Server #0 : 126.0

>> Energy Metrics:
- Total Consumption: 2799.999999999995
- Max Consumption: 22.22222222222222
- Average Consumption: 2799.999999999995
- Consumption per Server:
Server #0 : 2799.999999999995

#####
```

Listing 8: Metrics for RMS on a single server

Comparison Table

Algorithm	Pros	Cons	Possible Uses
FIFO	<ul style="list-style-type: none"> • Easy to implement and debug • Corresponds to the expected behavior of many applications (e.g. printers, chains of tasks such as piped commands, etc.) 	<ul style="list-style-type: none"> • Risks of starvation if a task is blocked • Risks of heavy tardiness if a task takes an exceptionally long amount of time to complete 	<ul style="list-style-type: none"> • When the application requires tasks to execute in a consecutive order (could be made safer with a timeout mechanic)
Round Robin	<ul style="list-style-type: none"> • Prevents starvation • Still rather easy to implement and debug 	<ul style="list-style-type: none"> • Tends to even more increase the number of deadline misses and average tardiness of tasks if the quantum is short 	<ul style="list-style-type: none"> • With a reasonable quantum and if a strict sequence in the order of treated tasks isn't needed as in FIFO, it can be a fair way to run tasks while being starvation-free.
EDF	<ul style="list-style-type: none"> • Lowest tardiness of tasks (but not necessarily lowest number of deadline misses) 	<ul style="list-style-type: none"> • A bit more complex to implement and maintain • Not starvation free as if no new arrival with higher priority comes then priorities are fixed and a blocked running task can block all the other tasks 	<ul style="list-style-type: none"> • Useful if we want highly responsive applications
RMS	<ul style="list-style-type: none"> • Lowest number of deadline misses 	<ul style="list-style-type: none"> • A bit more complex to implement and maintain (but once you have implemented EDF it's rather straightforward and reciprocally) • Not starvation free for the same reasons than with EDF 	<ul style="list-style-type: none"> • Probably the best algorithm for hard real-time operating systems where deadline misses isn't an option

Part II

Multiserver Scheduling

EDF

6.1 Implementation

6.2 Scheduling results

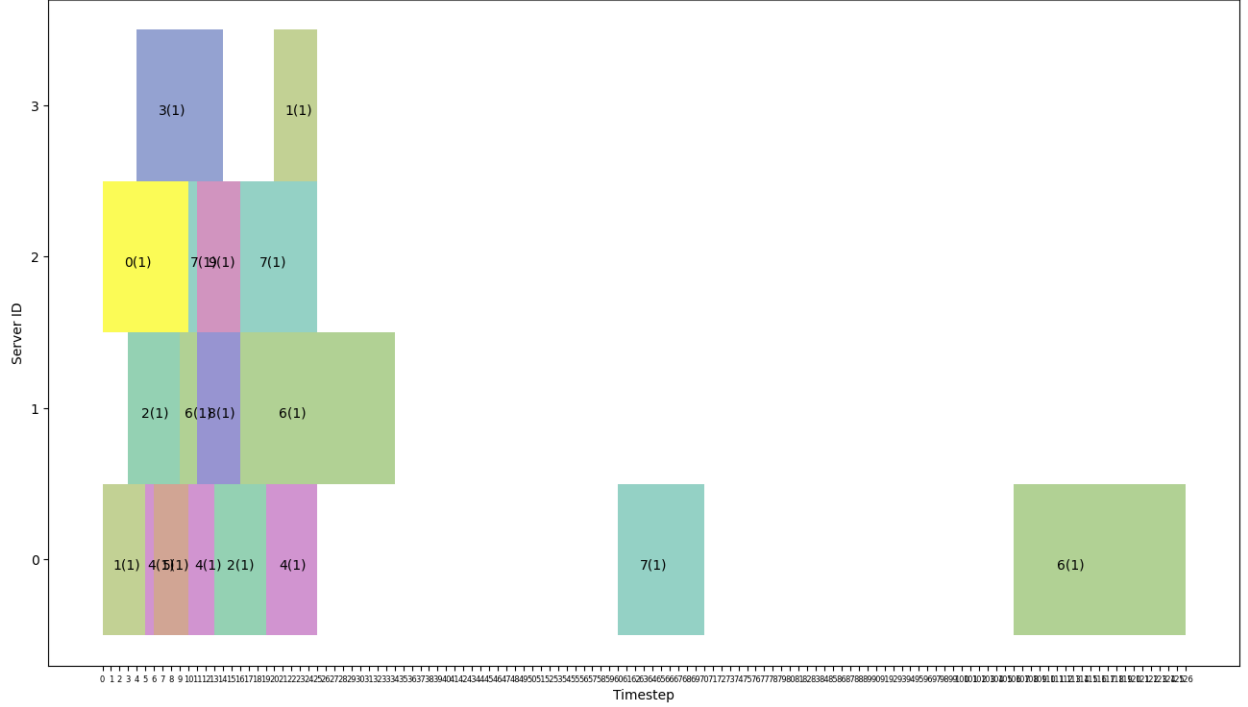


Figure 5: Output schedule produced by the EDF scheduling algorithm on multiple servers

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

1	0	0.0	5.0	1.0
4	0	5.0	6.0	1.0
2	1	3.0	9.0	1.0
5	0	6.0	10.0	1.0
0	2	0.0	10.0	1.0
6	1	9.0	11.0	1.0
7	2	10.0	11.0	1.0
4	0	10.0	13.0	1.0
3	3	4.0	14.0	1.0
8	1	11.0	16.0	1.0
9	2	11.0	16.0	1.0
2	0	13.0	19.0	1.0
4	0	19.0	25.0	1.0
7	2	16.0	25.0	1.0
1	3	20.0	25.0	1.0
6	1	16.0	34.0	1.0
7	0	60.0	70.0	1.0
6	0	106.0	126.0	1.0

Listing 9: Detail of the scheduling result

6.3 Metrics

```
##### SCHEDULE METRICS #####  
  
>> Scheduling Metrics:  
- Total Makespan: 126.0  
- Nb Deadline Misses: 0  
- Max Tardiness: 0.0  
- Average Tardiness: 0.0  
- Late Jobs:  
  
>> Servers Metrics:  
- Servers work load:  
Server #0 : 55.0  
Server #1 : 31.0  
Server #2 : 25.0  
Server #3 : 15.0  
  
>> Energy Metrics:  
- Total Consumption: 4077.777777777775  
- Max Consumption: 144.44444444444446  
- Average Consumption: 1019.4444444444438  
- Consumption per Server:  
Server #0 : 1222.2222222222208  
Server #1 : 1550.0  
Server #2 : 555.55555555555555  
Server #3 : 750.0  
  
#####
```

Listing 10: Metrics for EDF on multiple servers

RMS

7.1 Implementation

7.2 Scheduling results

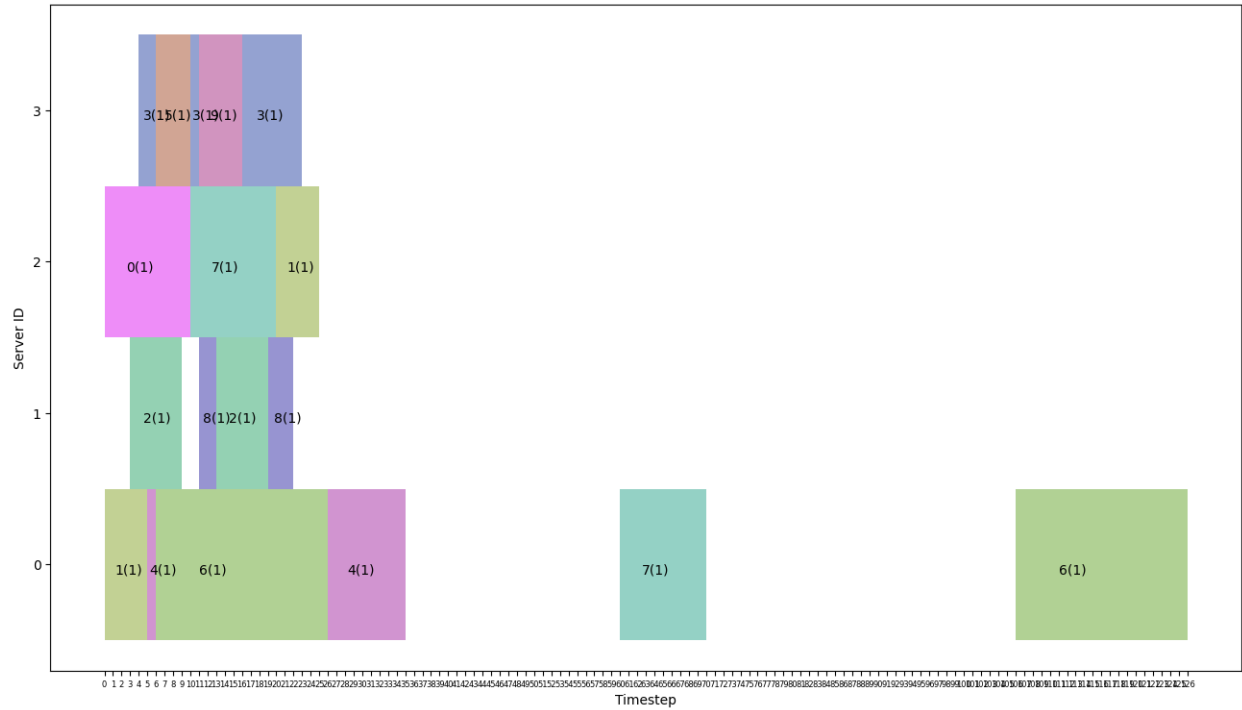


Figure 6: Output schedule produced by the RMS scheduling algorithm on multiple servers

Which corresponds to following file (`<jobID, serverID, startDate, endDate, frequency>`):

1	0	0.0	5.0	1.0
4	0	5.0	6.0	1.0
3	3	4.0	6.0	1.0
2	1	3.0	9.0	1.0
0	2	0.0	10.0	1.0
5	3	6.0	10.0	1.0
3	3	10.0	11.0	1.0
8	1	11.0	13.0	1.0
9	3	11.0	16.0	1.0
2	1	13.0	19.0	1.0
7	2	10.0	20.0	1.0
8	1	19.0	22.0	1.0
3	3	16.0	23.0	1.0
1	2	20.0	25.0	1.0
6	0	6.0	26.0	1.0
4	0	26.0	35.0	1.0
7	0	60.0	70.0	1.0
6	0	106.0	126.0	1.0

Listing 11: Detail of the scheduling result

7.3 Metrics

```
##### SCHEDULE METRICS #####  
  
>> Scheduling Metrics:  
- Total Makespan: 126.0  
- Nb Deadline Misses: 1  
- Max Tardiness: 6.0  
- Average Tardiness: 6.0  
- Late Jobs:  
[Job: J8{11.0a/0.0u/5.0rd/16.0ad/0.0p} | Tardiness: 6.0 ]  
  
>> Servers Metrics:  
- Servers work load:  
Server #0 : 65.0  
Server #1 : 17.0  
Server #2 : 25.0  
Server #3 : 19.0  
  
>> Energy Metrics:  
- Total Consumption: 3799.999999999997  
- Max Consumption: 144.44444444444446  
- Average Consumption: 949.9999999999992  
- Consumption per Server:  
Server #0 : 1444.44444444444425  
Server #1 : 850.0  
Server #2 : 555.55555555555555  
Server #3 : 950.0  
  
#####
```

Listing 12: Metrics for RMS on multiple servers

Comparison Table

Algorithm	Pros	Cons	Possible Uses
EDF	•	•	•
RMS	•	•	•

Part III

Multiserver Energy-aware Scheduling

FIFO

9.1 Implementation

9.2 Scheduling results

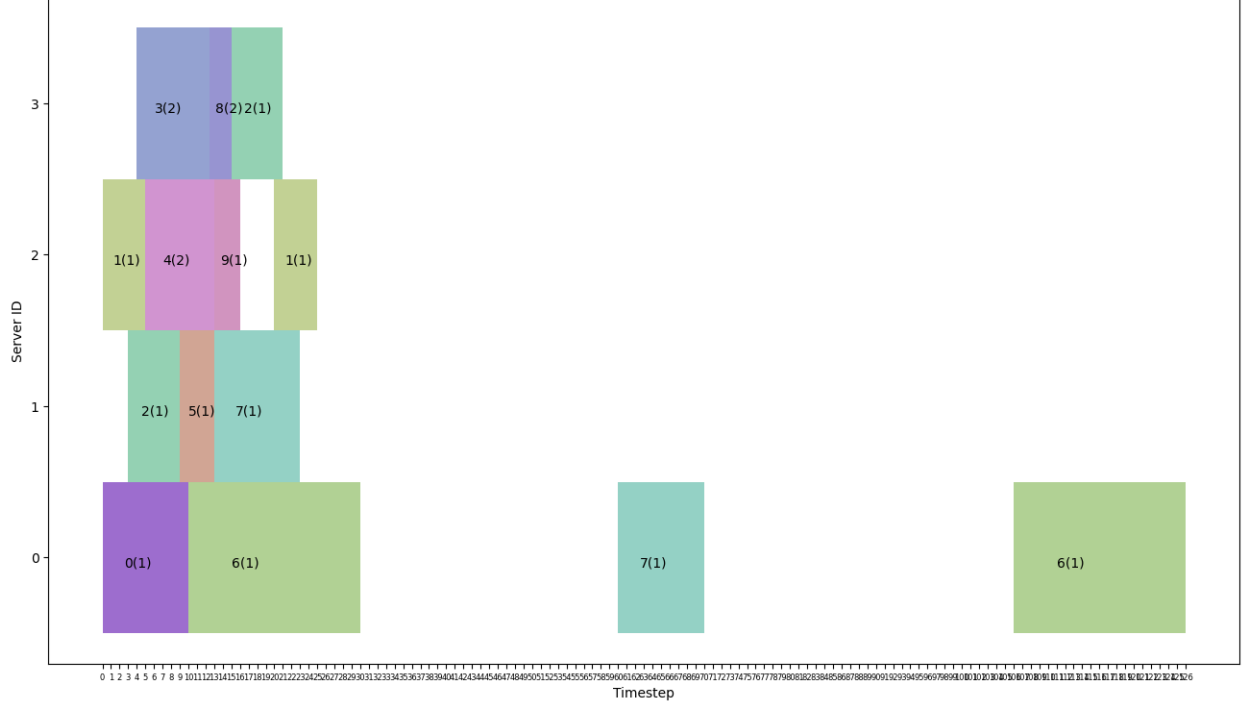


Figure 7: Output schedule produced by the FIFO scheduling algorithm on multiple servers and trying to use the smallest amount of power (with a power cap) while trying to not miss any deadline

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

1	2	0.0	5.0	1.0
2	1	3.0	9.0	1.0
0	0	0.0	10.0	1.0
3	3	4.0	12.5	2.0
5	1	9.0	13.0	1.0
4	2	5.0	13.0	2.0
8	3	12.5	15.0	2.0
9	2	13.0	16.0	1.0
2	3	15.0	21.0	1.0
7	1	13.0	23.0	1.0
1	2	20.0	25.0	1.0
6	0	10.0	30.0	1.0
7	0	60.0	70.0	1.0
6	0	106.0	126.0	1.0

Listing 13: Detail of the scheduling result

9.3 Metrics

```
##### SCHEDULE METRICS #####  
  
>> Scheduling Metrics:  
- Total Makespan: 126.0  
- Nb Deadline Misses: 0  
- Max Tardiness: 0.0  
- Average Tardiness: 0.0  
- Late Jobs:  
  
>> Servers Metrics:  
- Servers work load:  
Server #0 : 60.0  
Server #1 : 20.0  
Server #2 : 21.0  
Server #3 : 17.0  
  
>> Energy Metrics:  
- Total Consumption: 5833.3333333333485  
- Max Consumption: 361.11111111111114  
- Average Consumption: 1458.3333333333371  
- Consumption per Server:  
Server #0 : 1333.3333333333317  
Server #1 : 1000.0  
Server #2 : 999.9999999999998  
Server #3 : 2500.0  
  
#####
```

Listing 14: Metrics for FIFO on multiple energy aware servers

EDF

10.1 Implementation

10.2 Scheduling results

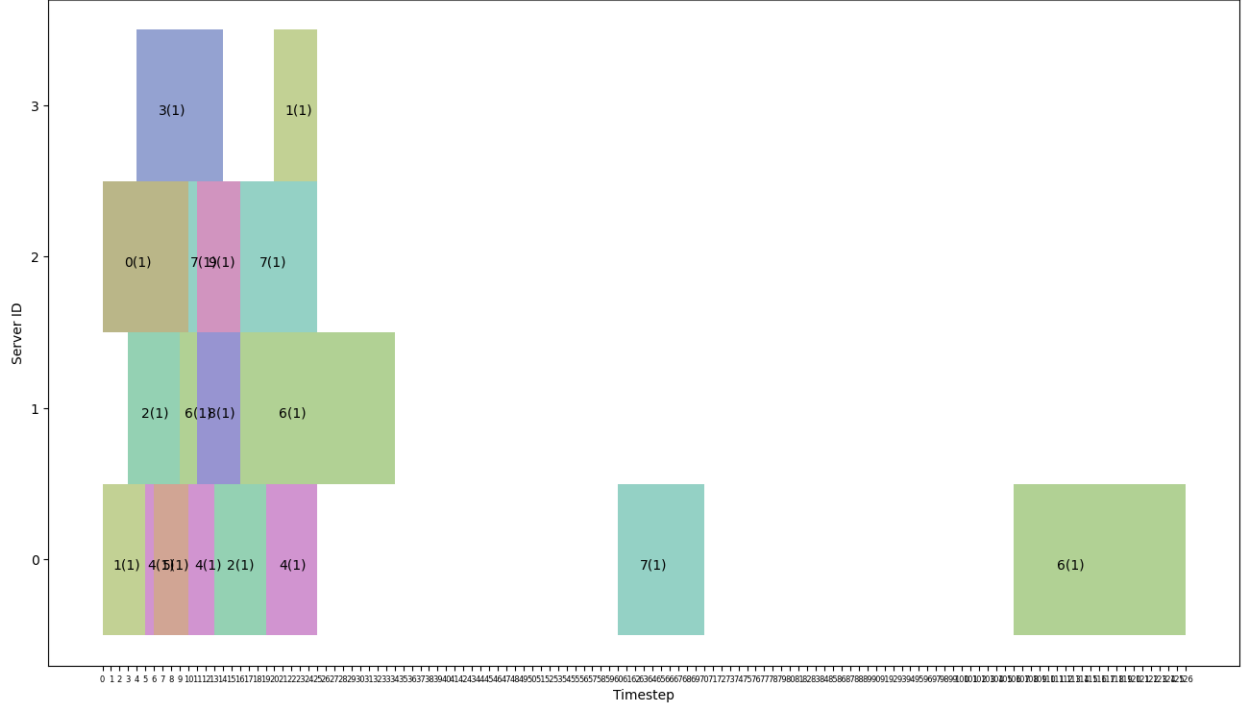


Figure 8: Output schedule produced by the EDF scheduling algorithm on multiple servers and trying to use the smallest amount of power (with a power cap) while trying to not miss any deadline

Which corresponds to following file (<jobID, serverID, startDate, endDate, frequency>):

1	0	0.0	5.0	1.0
4	0	5.0	6.0	1.0
2	1	3.0	9.0	1.0
5	0	6.0	10.0	1.0
0	2	0.0	10.0	1.0
6	1	9.0	11.0	1.0
7	2	10.0	11.0	1.0
4	0	10.0	13.0	1.0
3	3	4.0	14.0	1.0
8	1	11.0	16.0	1.0
9	2	11.0	16.0	1.0
2	0	13.0	19.0	1.0
4	0	19.0	25.0	1.0
7	2	16.0	25.0	1.0
1	3	20.0	25.0	1.0
6	1	16.0	34.0	1.0
7	0	60.0	70.0	1.0
6	0	106.0	126.0	1.0

Listing 15: Detail of the scheduling result

10.3 Metrics

```
##### SCHEDULE METRICS #####  
  
>> Scheduling Metrics:  
- Total Makespan: 126.0  
- Nb Deadline Misses: 0  
- Max Tardiness: 0.0  
- Average Tardiness: 0.0  
- Late Jobs:  
  
>> Servers Metrics:  
- Servers work load:  
Server #0 : 55.0  
Server #1 : 31.0  
Server #2 : 25.0  
Server #3 : 15.0  
  
>> Energy Metrics:  
- Total Consumption: 4077.777777777775  
- Max Consumption: 144.44444444444446  
- Average Consumption: 1019.4444444444438  
- Consumption per Server:  
Server #0 : 1222.2222222222208  
Server #1 : 1550.0  
Server #2 : 555.55555555555555  
Server #3 : 750.0  
  
#####
```

Listing 16: Metrics for EDF on multiple energy aware servers

Comparison Table

Algorithm	Pros	Cons	Possible Uses
FIFO	•	•	•
EDF	•	•	•