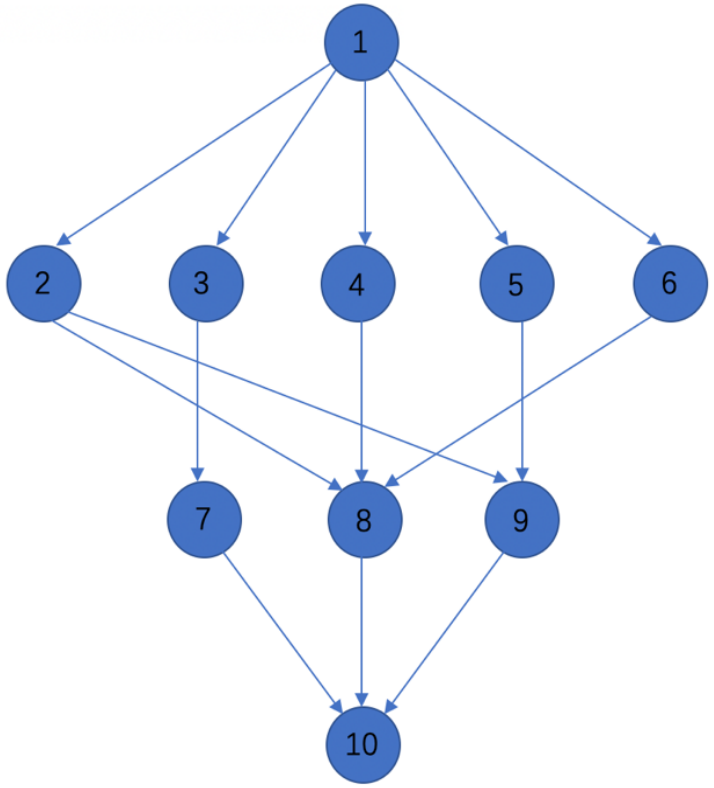
Northeastern University

EECE 7205 Fundamental of Computer Engineering

Project 2

Dongqi Xu ID:001342932

Part 1 :Original Input



Task	Core1	Core2	Core3
1	9	7	5
2	8	6	5
3	6	5	4
4	7	5	3
5	5	4	2
6	7	6	4
7	8	5	3
8	6	4	2
9	5	3	2
10	7	4	2

First we use the example in the paper .

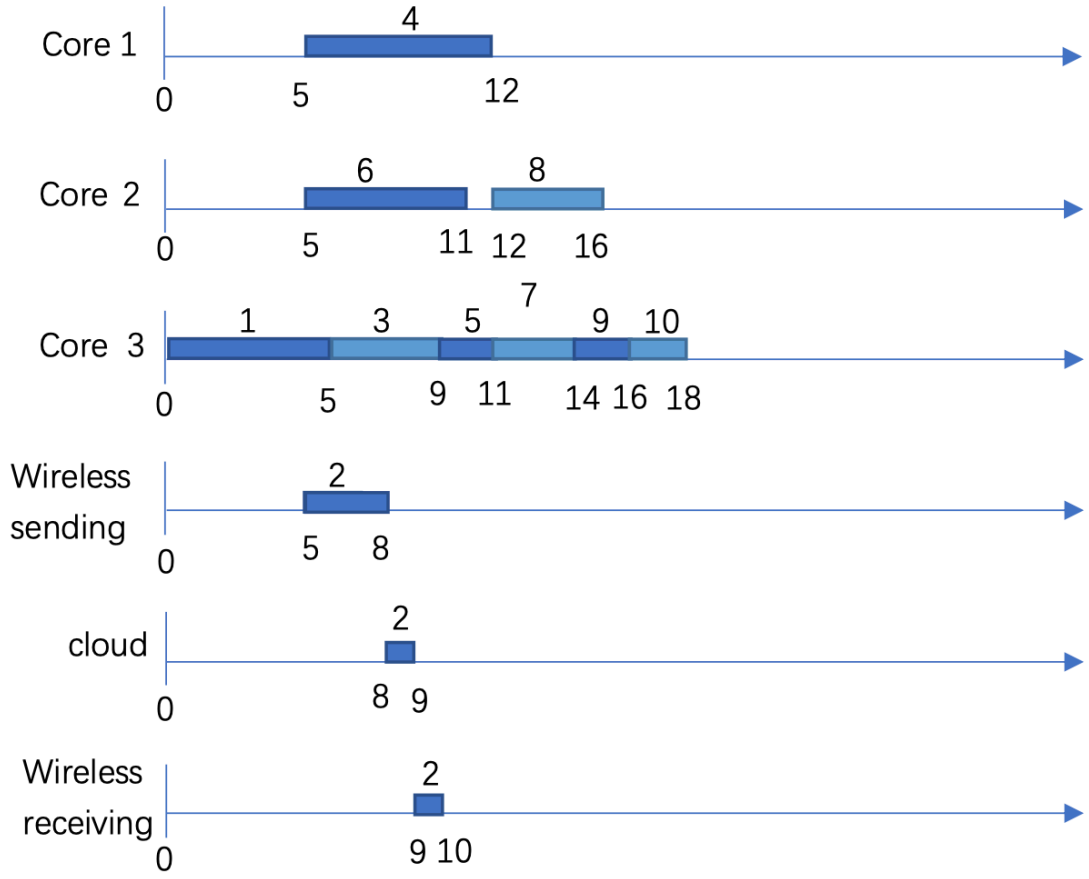
Task Number = 10
Core Number = 3

$T_i^s = 3;$
$T_i^c = 1;$ $(1 \leq i \leq N)$
$T_i^r = 1;$

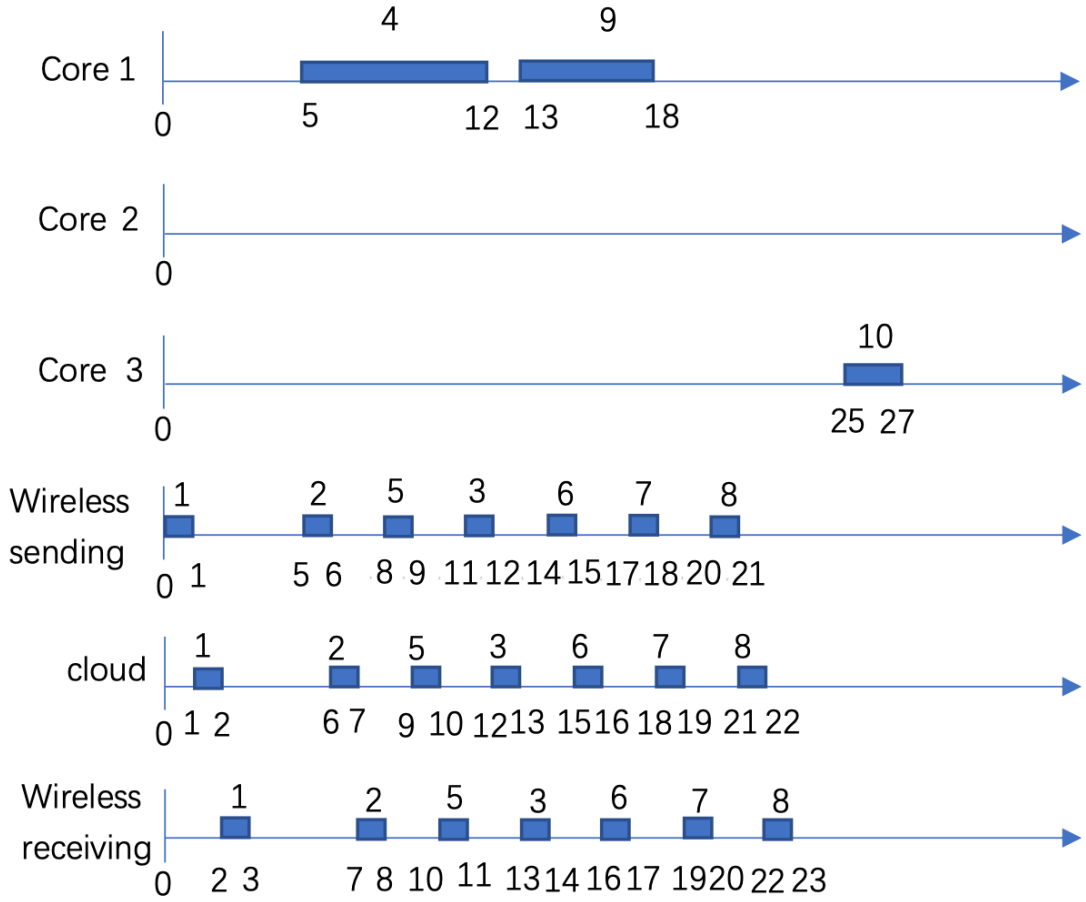
$P^s = 0.5;$
$P_1 = 1;$
$P_2 = 2;$
$P_3 = 4;$

Part 1 :Original Output

Initial Scheduling:



MCC scheduling:

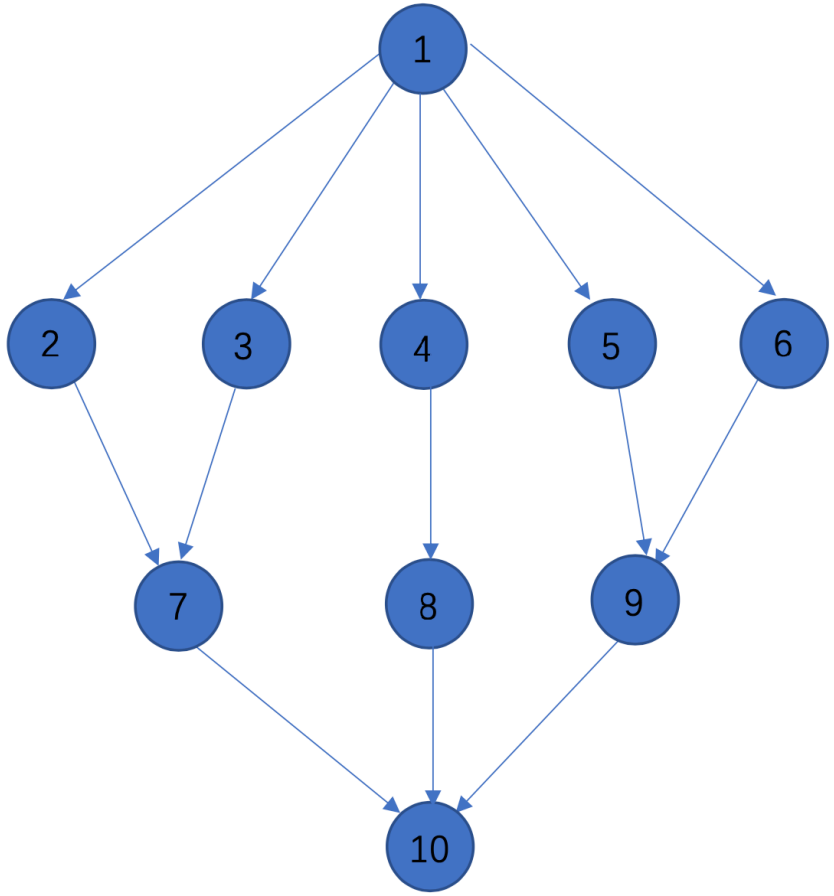


Part 1 : Result of Code

```
Task scheduling result by the initial task scheduling algorithm:
Core1: 5–12 for Task4;
Core2: 5–11 for Task6; 13–16 for Task8;
Core3: 0–5 for Task1; 5–9 for Task3; 9–11 for Task5; 11–14 for Task7; 14–16 for Task9; 16–18 for Task
10;
wireless sending : 5–8 for Task2;
Cloud: 8–9 for Task2;
wireless receiving : 9–10 for Task2;

Energy Consumption of initial scheduling: 100.5   Total Completion Time: 18


Task scheduling result by the MCC task scheduling algorithm:
Core 1: 5–12 for Task4; 13–18 for Task9;
Core 2:
Core 3: 25–27 for Task10;
wireless sending : 0–1 for Task1; 5–6 for Task2; 8–9 for Task5; 11–12 for Task3; 14–15 for Task6; 17–
18 for Task7; 20–21 for Task8;
Cloud: 1–2 for Task1; 6–7 for Task2; 9–10 for Task5; 12–13 for Task3; 15–16 for Task6; 18–19 for Task
7; 21–22 for Task8;
wireless receiving : 2–3 for Task1; 7–8 for Task2; 10–11 for Task5; 13–14 for Task3; 16–17 for Task6;
 19–20 for Task7; 22–23 for Task8;

Energy Consumption of scheduling: 27   Total Completion Time: 27
```

Energy consumption is 27 , and the time cost is 27. So the result satisfies the time constraint.

Part 2 :New Input



Task	Core1	Core2	Core3
1	5	7	9
2	5	6	8
3	4	5	6
4	3	5	7
5	2	4	5
6	4	6	7
7	3	5	8
8	2	4	6
9	2	3	5
10	2	4	7

We choose another the example to test the code.

Task Number = 10
Core Number = 3

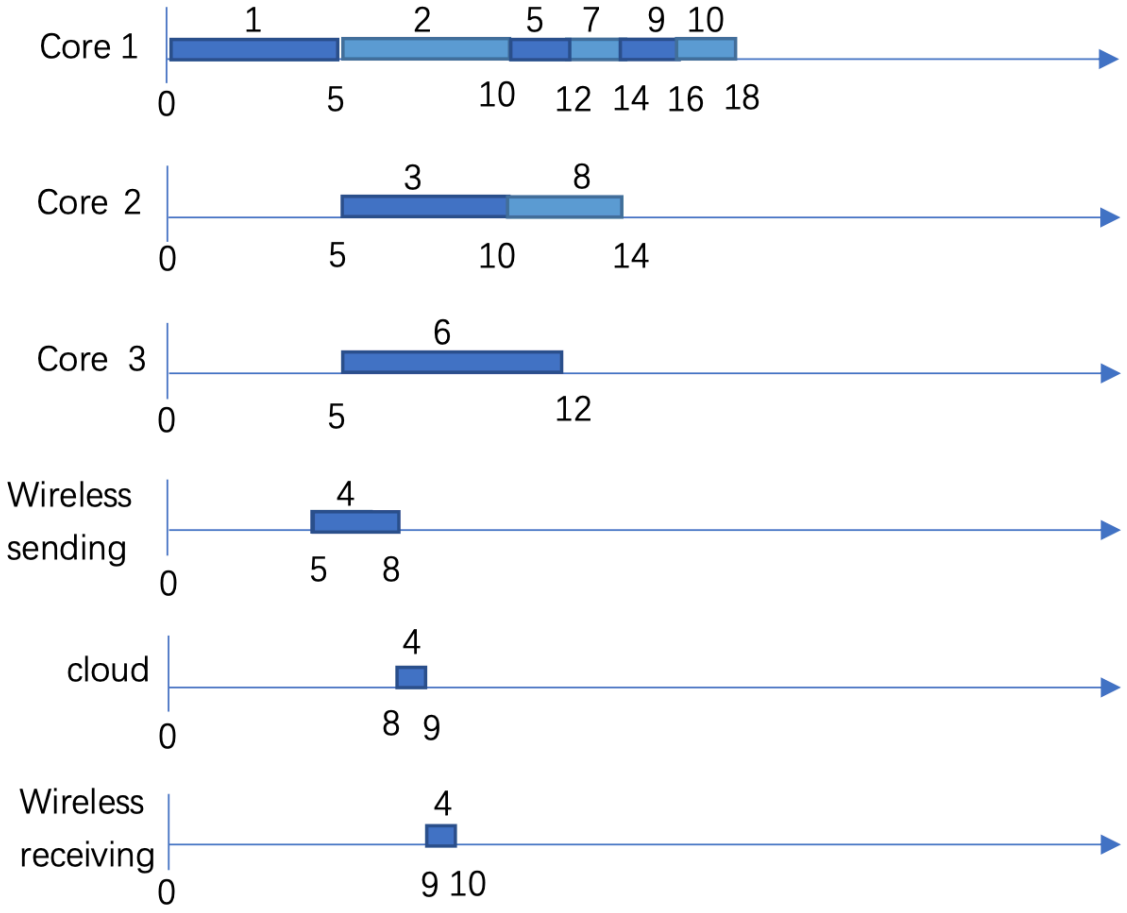
T_i^s = 3;
T_i^c = 1;
T_i^r = 1;

(1 ≤ i ≤ N)

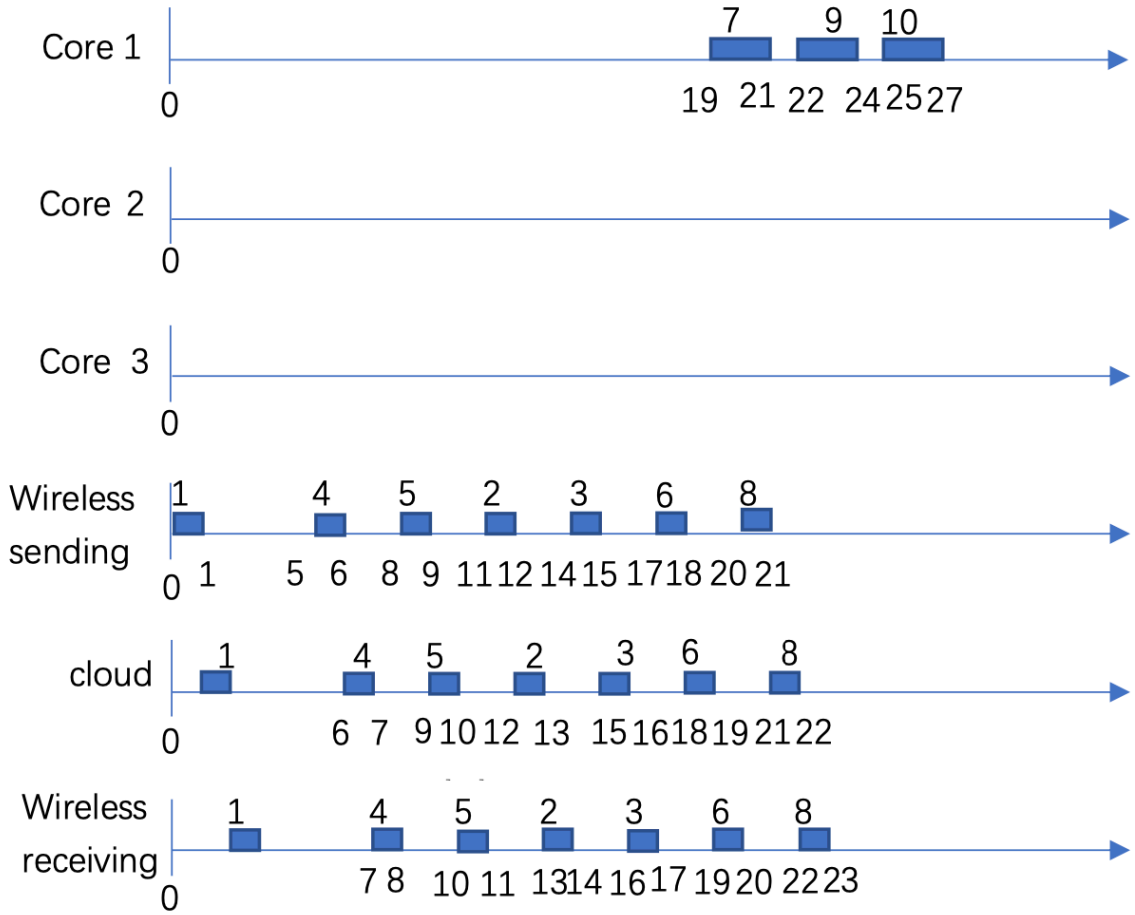
P^s = 0.5;
P_1 = 1;
P_2 = 2;
P_3 = 4;

Part 2 :New Output

Initial Scheduling:



MCC scheduling:



Part 2 : Result of Code

```
Task scheduling result by the initial task scheduling algorithm:
Core1: 0-5 for Task1; 5-10 for Task2; 10-12 for Task5; 12-14 for Task7; 14-16 for Task9; 16-18 for
 Task10;
Core2: 5-10 for Task3; 10-14 for Task8;
Core3: 5-12 for Task6;
wireless sending : 5-8 for Task4;
Cloud: 8-9 for Task4;
wireless receiving : 9-10 for Task4;

Energy Consumption of initial scheduling: 65.5   Total Completion Time: 18


Task scheduling result by the MCC task scheduling algorithm:
Core 1: 19-21 for Task7; 22-24 for Task9; 25-27 for Task10;
Core 2:
Core 3:
wireless sending : 0-1 for Task1; 5-6 for Task4; 8-9 for Task5; 11-12 for Task2; 14-15 for Task3;
17-18 for Task6; 20-21 for Task8;
Cloud: 1-2 for Task1; 6-7 for Task4; 9-10 for Task5; 12-13 for Task2; 15-16 for Task3; 18-19 for T
ask6; 21-22 for Task8;
wireless receiving : 2-3 for Task1; 7-8 for Task4; 10-11 for Task5; 13-14 for Task2; 16-17 for Tas
k3; 19-20 for Task6; 22-23 for Task8;

Energy Consumption of scheduling: 13   Total Completion Time: 27
```

Energy consumption is 13 , and the time cost is 27. So the result satisfies the time constraint.

Code：

- I write the code using the main idea and frame work in the paper , and make some references on the website for some details.
- The code is written exactly according to the steps in the paper：
- Step 1 :
- 1. primary assignment：Compare the time of each task used in the core and cloud , determine where the task should be.
- 2. task prioritizing : Make the task priority and number to be a pair, and calculate the priority of the tasks.
- 3. execution unit selection : Generate initial scheduling.

Code ：

- Step 2 :
- 1.Outer loop : Determine which task need to be moved to another core or to the cloud.
- 2.Kernael : Move the tasks chosen to be on other cores or on the cloud to generate the MCC scheduling.

Responsibility ：

- All works have been done by Dongqi Xu.