

## Exercise 4. Performance Evaluation

1. Based on the following data decomposition strategy, fill in the table below for  $n = 100$  and  $p = 7$

- First element controlled by process  $i = [i*n/p]$
- Last element controlled by process  $i = [(i+1)*n/p]-1$
- Block size = Last - First + 1

Process	First Element	Last Element	Block Size
0	0	13	14
1	14	27	14
2	28	41	14
3	42	56	15
4	57	70	15
5	71	84	14
6	85	99	15

Total = 100

โจทย์กำหนดจำนวนตัวเลข  $n$  มาให้ 100 ตัว และ  $p$  หรือ process มาให้ทั้งหมด 7 ตัว (0-6)

### วิธีคิดโปรเซสตัวที่ 0

ค่าตัวแรก หรือ First Element( $i$ ) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 0,  $i = (0100)/7$  จะได้ 0 ค่าตัวสุดท้าย หรือ Last Element( $i$ ) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง  
 $i$  ที่ 0,  $i = [(0+1)*100/7]-1$  ค่าที่ได้คือ 13.285714286 ซึ่งเราจะต้องปัดทศนิยมลงจะได้ค่าเท่ากับ 13

Block Size หรือ ช่องเมมโมรี สูตรคือ ค่าสุดท้าย - ค่าแรก + 1

เช่น Block Size ที่ โปรเซส 0 มีค่าเป็น  $13 - 0 + 1$  เท่ากับ 14

### วิธีคิดโปรเซสตัวที่ 1

ค่าตัวแรก หรือ First Element( $i$ ) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 1,  $i = (1100)/7$  จะได้ 14.285714286

ซึ่งเราจะต้องปัดทศนิยมลงจะได้ค่าเท่ากับ 14 ค่าตัวสุดท้าย หรือ Last Element( $i$ ) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 1,  $i = [(1+1)*100/7]-1$  ค่าที่ได้คือ 27.571428571 ซึ่งเราจะต้องปัดทศนิยมลงจะได้ค่าเท่ากับ 27

Block Size ที่ โปรเซส 1 มีค่าเป็น  $27 - 14 + 1$  เท่ากับ 14

### วิธีคิดโปรเซสตัวที่ 2

ค่าตัวแรก หรือ First Element(i) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 2 ,  $i = (2100)/7$  จะได้ 28

ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 28 ค่าตัวสุดท้าย หรือ Last Element(i) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 2 ,  $i = [(2+1)*100/7]-1$  ค่าที่ได้คือ 41.857142857 ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 41

Block Size ที่ โปรเซส 0 มีค่าเป็น  $41 - 28 + 1$  เท่ากับ 14

### วิธีคิดโปรเซสตัวที่ 3

ค่าตัวแรก หรือ First Element(i) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 3 ,  $i = (3100)/7$  จะได้ 42.857142857

ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 42 ค่าตัวสุดท้าย หรือ Last Element(i) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 3 ,  $i = [(3+1)*100/7]-1$  ค่าที่ได้คือ 56.142857143 ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 56

Block Size ที่ โปรเซส 0 มีค่าเป็น  $56 - 42 + 1$  เท่ากับ 14

### วิธีคิดโปรเซสตัวที่ 4

ค่าตัวแรก หรือ First Element(i) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 4 ,  $i = (4100)/7$  จะได้ 57.142857143

ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 57 ค่าตัวสุดท้าย หรือ Last Element(i) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 4 ,  $i = [(4+1)*100/7]-1$  ค่าที่ได้คือ 70.428571429 ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 70

Block Size ที่ โปรเซส 0 มีค่าเป็น  $70 - 57 + 1$  เท่ากับ 15

### วิธีคิดโปรเซสตัวที่ 5

ค่าตัวแรก หรือ First Element(i) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 5 ,  $i = (5100)/7$  จะได้ 71.428571429

ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 71 ค่าตัวสุดท้าย หรือ Last Element(i) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 5 ,  $i = [(5+1)*100/7]-1$  ค่าที่ได้คือ 84.714285714 ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 84

Block Size ที่ โปรเซส 0 มีค่าเป็น  $71 - 84 + 1$  เท่ากับ 15

### วิธีคิดโปรเซสตัวที่ 6

ค่าตัวแรก หรือ First Element(i) : สูตรคือ  $i = (in)/p$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 6 ,  $i = (6100)/7$  จะได้ 85.714285714

ซึ่งเราจะต้องบดทศนิยมลงจะได้ค่าเท่ากับ 85 ค่าตัวสุดท้าย หรือ Last Element(i) : สูตรคือ  $i = [(i+1)*n/p]-1$  ซึ่ง  $i$  คือ โปรเซสในตาราง

$i$  ที่ 6 ,  $i = [(6+1)*100/7]-1$  ค่าที่ได้คือ 99

Block Size ที่ โปรเซส 0 มีค่าเป็น  $85 - 99 + 1$  เท่ากับ 15

## 2. Suppose that a parallel program will be executed on 16 processors.

- 1.How much the ideal speedup would achieve?

- 2.If the program has a 4% serial portion, how much speedup can it achieve based on Amdahl's Law?
- 3.How much speedup limit the program can expect?

ข้อแรกตอบ 16 ตัว

ข้อสองตอบ Speedup =  $1 / (0.04 + (1-0.04)/16) = 10$  จากสูตร

Speedup(p) =  $1/f + (1 - f) / p$  # p คือจำนวนโปรเซส ส่วน f หรือ fraction คือเปอร์เซ็นต์การทำงานโปรแกรม (Serial Portion)

ตัวอย่างเช่น ถ้าการทำงานที่สามารถทำแบบขนานได้คิดเป็น 11% (0.11)

และการทำงานแบบลำดับคิดเป็น 89% (0.89)จะได้ค่า Speedup =  $1 / ((1-0.11) + (0.11/0.89)) = 0.986586853$

ข้อสามตอบ  $1/(0.04+(1-0.04))/16 = 10$

ตามสูตร

limit(p -> infinity) =  $1/f + (1 - f) / p$  # p คือจำนวนโปรเซส ส่วน f คือเปอร์เซ็นต์การทำงานโปรแกรม (Serial Portion)

3. Given a scaled speedup of 20 on 32 processors, how much serial fraction of the program is according the Gustafson's Law

สูตรคือ

# Gustafson's Law

Scaled speedup(pเก่า) =  $p_{ใหม่} + (1 - p_{ใหม่}) * s$  # p คือจำนวนโปรเซสเซอร์

ต้องเอาโปรเซสเก่ามาคิดด้วย

20 จาก p เก่า = 32 + (1-32)s

s(31) = 32 - 20

s = 32 - 20 / 31

s = 0.387096774 หรือ 38.7 %

4. A program attains 89% efficiency with a serial fraction of 2%. Approximately how many processors are being used according to Amdahl's law?

$0 \leq E(p) \leq 1$  # E(p) ที่ p เป็น 0.89

Speedup(p) =  $1/f + (1 - f) / p$  # สูตรที่ 1

# จะหา Efficiency ยังไง

$E(p) = \text{Speedup}(p) / p$

คำตอบคือ 7.1797 แต่ต้องปัดขึ้นเป็น 8

$$E(p) = \frac{S(p)}{p} E(p) = \frac{1}{0.02 + (1-0.02)*p}$$

$$E(p) = \frac{1}{0.02 + (0.98)*p}$$

$$E(p) = \frac{\frac{1}{0.02 + (0.98)*p}}{p}$$

$$E(p) = \frac{1}{0.02 + (0.98/p)} * \frac{1}{p}$$

$$0.89 = \frac{1}{0.02 + (\frac{0.98}{p})} * \frac{1}{p}$$

$$0.89 = \frac{1}{(0.02 + \frac{0.98}{p}) * p}$$

$$0.89 * (0.02 + \frac{0.98}{p}) * p = 1$$

$$0.89p * (0.02 + \frac{0.98}{p}) = 1$$

$$0.0178p + 0.8722 = 1$$

$$0.0178p = 1 - 0.8722$$

$$p = \frac{1 - 0.8722}{0.0178}$$

$$p = \frac{0.1278}{0.0178}$$

$$p = 7.1797$$

$$p \approx 8$$

วิธีคิด

```
E(p) = Speedup(p) / p
# Speedup(p) หรือ S(p) จะต้องใช้ตามกฎที่ให้มา Amdahl หรือ Gustafson
E(p) = 1 / (f + ( 1 - f ) / p ) / p
# แทนค่า
0.89 = [ 1 / 0.02 + (( 1 - 0.02 ) / p ) ] / p
0.89 = [ 1 / 0.02 + (( 0.98 ) / p ) ] / p # คิดค่าในวงเล็บให้หมด
0.89 = [ 1 / 0.02 + ( 0.98 / p ) ] / p # ย้ายข้างสมการ
```

```
0.89 * p * ( 0.02 + ( 0.98 / p )) = 1
0.0178p + ( 0.8722p / p ) = 1
0.0178p + 0.8722 = 1
0.0178p = 1 - 0.8722
0.0178p = 0.1278
p = 0.1278 / 0.0178
p = 7.1797 # เราจะต้องปัดค่าขึ้น
p = 8
```

5. Modify the IntegerSum program to measure the execution time of the program by using the MPI\_Wtime(). Compare the parallel execution time with a sequential version and discuss the result.

When run Parallel program, this program is faster than Sequential Program

Because, Sequential Program need to wait other process to get job done and send back to root node

คำตอบคือ เมื่อเราทำการรันโปรแกรมแบบขนาน(Parallel program)เวลาในการรันจะเร็วกว่าการรันแบบเป็น

ลำดับ(Sequential program)

เพราะว่าการรันโค้ดแบบเป็นลำดับ(Sequential program)นั้นต้องรอตัวโปรเซสอื่นทำงานเสร็จก่อนและส่งข้อมูลกลับไป