

ข้อสังเกต

จากโจทย์และโค้ดที่ได้มาจากอาจารย์เมื่อทำการลองรันดูแล้วจะได้ผลลัพธ์อยู่ที่ 888701676 ซึ่งเป็นผลลัพธ์ที่ได้มาจากการเรียกใช้ฟังก์ชัน sum() ซ้ำอยู่ใน loop 1000 ล้านรอบ และมีเวลาในการหาผลลัพธ์อยู่ที่ประมาณ 20 วินาที สามารถตีความได้ว่าจากการทำงานของโปรแกรมมีการรันการทำงานอยู่บน thread เดียวซึ่งทำให้การทำงานมีเวลานาน ดังนั้นจึงคิดวิธีในการเพิ่มประสิทธิภาพได้เป็นการสร้าง thread มาเพิ่มเพื่อเป็นการแบ่งจำนวนงานการทำงานของแต่ละ thread และมีสมมติฐานว่าจะทำให้ความเร็วในการทำงานของโปรแกรมมีความเร็วขึ้น

Version 0

สังเกตพบว่า : มีการทำงาน loop 1000 ล้านรอบเพื่อคำนวณค่าที่อ่านได้จากไฟล์ ซึ่งเป็นการทำงานที่เยอะมากๆ ทำให้โปรแกรมมีการทำงานที่มีประสิทธิภาพที่แย่ หากนับจากเวลา

ปัญหา : มีการทำงานเป็น iteration ที่มีจำนวนเยอะมากๆ

วิธีแก้ : จะทดลองสร้าง Thread ย่อยลงมาเพื่อแบ่งการทำงานเป็น 4 Thread คือฟังก์ชัน Sum มา 4 อัน ที่มี algorithm เหมือนกับ function Sum ต้นฉบับ แต่ละ Sum จะมีใช้ index ต่างกันคือ ตั้งแต่ 0-250 ล้าน , 250-500 ล้าน, 500-750 ล้าน และ 750-1000 ล้าน เพื่อไว้สำหรับในการรันแต่ละ Thread

เนื่องจาก : -

ผลลัพธ์ : เวลาที่ใช้ Compile ลดลงประมาณ 10 วินาที

สรุปผลลัพธ์ :

รันครั้งที่ 1 6331ms

รันครั้งที่ 2 6273ms

รันครั้งที่ 3 6114ms

Version 1

สังเกตพบว่า : จาก Version 0 พบว่าวิธีที่ใช้ยังไม่ถูกต้อง เพราะไม่มีการใช้ฟังก์ชัน Sum ต้นฉบับที่อาจารย์กำหนดไว้

ปัญหา : โค้ดยังมีการทำงานด้วยโค้ดที่ไม่ตรงความต้องการ (requirement)

วิธีแก้ : สร้าง Thread เพิ่มขึ้นมา 4 Thread เพื่อใช้คำนวณฟังก์ชัน sum() พร้อมๆกัน

เนื่องจาก : วิธีแก้ไขของ Version 0 ยังไม่ถูกต้อง

ผลลัพธ์ : เวลาที่ใช้คอมไพล์เร็วขึ้นเล็กน้อย

สรุปผลลัพธ์ :

รันครั้งที่ 1 4340ms

รันครั้งที่ 2 4275ms

รันครั้งที่ 3 4325ms

Version 2

สังเกตพบว่า : ยังสามารถเพิ่มประสิทธิภาพการคำนวณได้อีก

ปัญหา : ทำอย่างไรถึงจะเพิ่มประสิทธิภาพการคำนวณให้เพิ่มขึ้น

วิธีแก้ : เพิ่ม Thread จาก 4 เป็น 8 Thread, จาก 8 เป็น 16, จาก 16 เป็น 32 และจาก 32 เป็น 64

เนื่องจาก : การเพิ่ม Thread เป็นการเพิ่มประสิทธิภาพในการคำนวณให้เพิ่มขึ้น ดังนั้นการเพิ่ม Thread จึงจะทำให้

เวลาการทำงานลดลง

ผลลัพธ์ : เวลาที่ใช้คอมไพล์เร็วขึ้นค่อนข้างมาก แต่เวลาจะคงที่เมื่ออยู่ที่ 16 Thread เนื่องจากขีดจำกัดด้านประสิทธิภาพของ Hardware

ผลลัพธ์ในเครื่องที่ 1

Number of thread	First time running (ms)	Second time running (ms)	Third time running (ms)	Average time (ms)
4	5649	5633	5467	5583
8	3159	3147	3176	3160.67
16	2691	3056	3055	2934
32	2636	3000	2800	2812
64	2781	2623	2699	2701

ผลลัพธ์ในเครื่องที่ 2

Number of thread	First time running (ms)	Second time running (ms)	Third time running (ms)	Average time (ms)
4	4340	4275	4325	4313.33
8	2553	2569	2521	2547.67
16	2042	2105	1975	2040.67
32	2132	2111	2022	2088.33
64	2094	2134	2037	2088.33

Food for thought (อาหารสมอง)

จะเกิดอะไรขึ้นกับประสิทธิภาพของโปรแกรมเมื่อมีการเพิ่ม thread ให้มากขึ้นเรื่อยๆ จะทำให้เวลาในการทำงานลดลงตลอดหรือไม่ แล้วทำไม???

Ans การเพิ่มจำนวน thread ไปเรื่อยๆจะเป็นการแบ่งงานทั้งหมด เป็นงานชิ้นเล็กๆเพื่อให้แต่ละ thread ช่วยกันทำงานพร้อมๆกัน แต่ถ้าจำนวน thread มีเกินจำนวนที่ hardware รับไหว ก็จะมีการทำงานของ thread ที่พร้อมๆกันเท่ากับจำนวนของ thread ที่ hardware มีเท่านั้น ดังนั้นการเพิ่มจำนวน thread ไปเรื่อยๆจะไม่ได้ทำให้การทำงานของโปรแกรมเร็วขึ้นเสมอไป

สมาชิก

63010035	กฤษฎาง	แสงศิริผล
63010082	กิตติภณ	สิงห์ชม
63010183	ชนน	กุลกัตติมาส
63010229	ชินกฤต	ปิ่นคล้าย
63010283	ณภัทร	มูลพินิจ
63011013	สุเมธ	สวนสำราญ