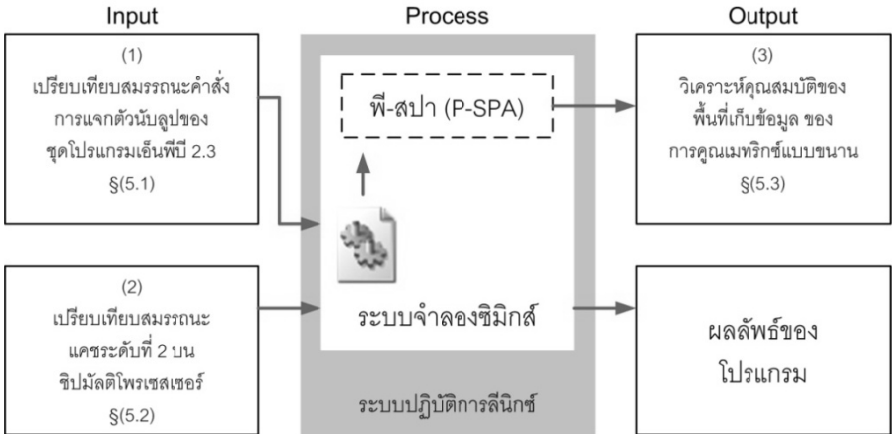
บทที่ 5

การทดลองเบื้องต้น

บทที่ผ่านมาสรุปขั้นตอนการวิจัยและพัฒนาเครื่องมือวิเคราะห์สมรรถนะพี-สปา เพื่อให้สำเร็จได้ตามวัตถุประสงค์ เนื้อหาในบทนี้อธิบายการทดลองเบื้องต้น ซึ่งดำเนินการในระหว่างการพัฒนาเครื่องมือวิเคราะห์พี-สปา เพื่อศึกษาวิธีการพัฒนาโมดูลพี-สปาให้ประสานงานกับระบบจำลองซิมิกส์ในการเก็บข้อมูลการใช้งานแคชได้อย่างถูกต้อง ซึ่งทุกการทดลองที่กล่าวถึงในบทนี้ ดำเนินการโดยใช้โมดูลพี-สปาที่ได้พัฒนาไปบางส่วนเป็นเครื่องมือ

ภาพที่ 5.1 ภาพรวมของการทดลองเบื้องต้น

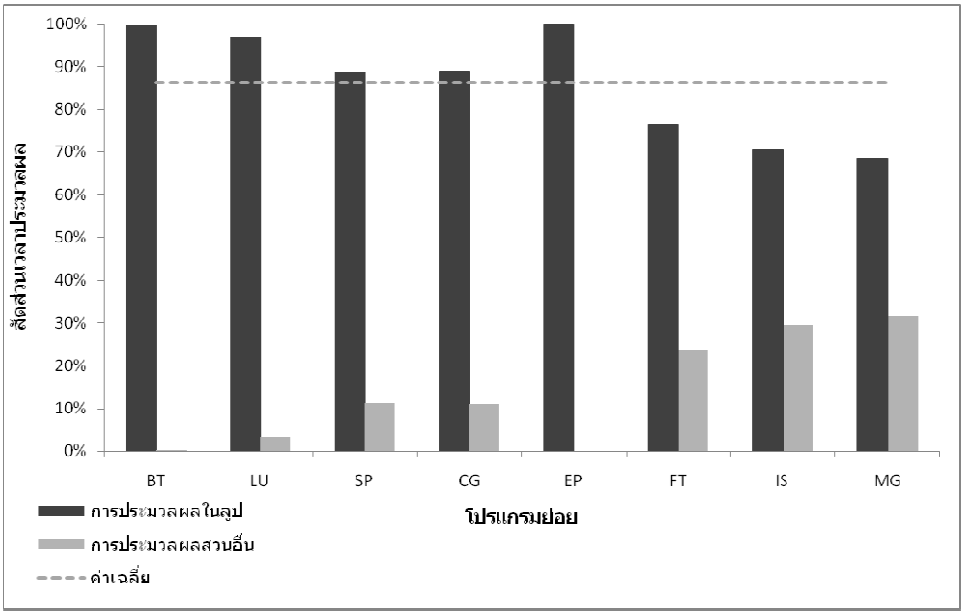


การทดลองเบื้องต้นแบ่งเป็นสามการทดลองตามจุดที่พี-สปาจะต้องประสานงานกับระบบจำลองซิมิกซ์ (ภาพที่ 5.1) ซึ่งประกอบด้วย (1) การทดลอง*เปรียบเทียบสมรรถนะคำสั่งการแจกตัวนับลูปของชุดโปรแกรมเอ็นพีบี 2.3* เพื่อศึกษาการนำโปรแกรมแบบโอเพนเอ็มพีเข้าไปประมวลผลบนระบบจำลองซิมิกส์ (2) การทดลอง*เปรียบเทียบสมรรถนะของแคชระดับที่ 2 บนชิปมัลติโพรเซสเซอร์* ที่มี 2 หน่วยประมวลผล เพื่อศึกษาวิธีการปรับค่าพารามิเตอร์และการประสานงานเก็บข้อมูลจากโมดูลจี-แคช และ (3) การทดลอง*วิเคราะห์คุณสมบัติการใช้พื้นที่เก็บข้อมูลของฟังก์ชันการคูณเมทริกซ์แบบขนาน* ที่ใช้จำนวนเทรดประมวลผลแตกต่างกัน เพื่อศึกษาวิธีการนำเสนอข้อมูลการใช้ประโยชน์ของแคชในรูปแบบของกราฟ เนื้อหาของบทนี้นำเสนอรายละเอียดเกี่ยวกับการทดลองตามลำดับ และในส่วนสุดท้ายแสดงการสรุปผลการศึกษาจากการทดลองเบื้องต้นทั้งหมด

5.1 การเปรียบเทียบสมรรถนะคำสั่งการแจกตัวนับลูปของชุดโปรแกรมเอ็นพีบี 2.3

ดังที่กล่าวในหัวข้อ 4.3.1 ในรายละเอียดว่าชุดโปรแกรมเอ็นพีบี 2.3 ขององค์การนาซ่าเป็นโปรแกรมที่พัฒนาด้วยโอเพนเอ็มพี บนภาษาซี ประกอบด้วยแปดโปรแกรมย่อย ได้แก่ บีที แอลยู และเอสพี ซึ่งเป็นโปรแกรมจำลองการไหลของสารเคมี และซีจี อีพี เอฟที ไอเอส และเอ็มจี ซึ่งเป็นภาระงานหลัก (Kernel) ของโปรแกรมการคำนวณทางวิทยาศาสตร์ ชุดโปรแกรมเอ็นพีบีมีการประมวลผลในลูปเป็นส่วนประมวลผลหลักของโปรแกรม ซึ่งจากภาพที่ 5.2 ที่แสดงสัดส่วนเวลาประมวลผลของชุดโปรแกรมเอ็นพีบี พบว่า ค่าเฉลี่ยเวลาการประมวลผลในลูปของทั้งแปดโปรแกรมสูงถึงร้อยละ 86 การเลือกคำสั่งการแจกตัวนับลูปที่เหมาะสมจะช่วยให้โปรแกรมมีสมรรถนะการประมวลผลสูงขึ้นได้ อย่างไรก็ตาม แต่ละโปรแกรมก็มีคำสั่งการแจกตัวนับลูปที่เหมาะสมแตกต่างกัน ดังนั้น การทดลองนี้จึงมีเป้าหมายสองประการ คือ (1) เพื่อเปรียบเทียบสมรรถนะคำสั่งการแจกตัวนับลูปของแต่ละโปรแกรมย่อย ในชุดโปรแกรมเอ็นพีบี 2.3 และ (2) เพื่อศึกษาการประมวลผลชุดโปรแกรมเอ็นพีบี 2.3 บนระบบจำลองซิมิกส์ ซึ่งโครงงานวิจัยนี้จะนำโปรแกรมดังกล่าวมาใช้ทดสอบความถูกต้องของพี-สปาในภายหลัง

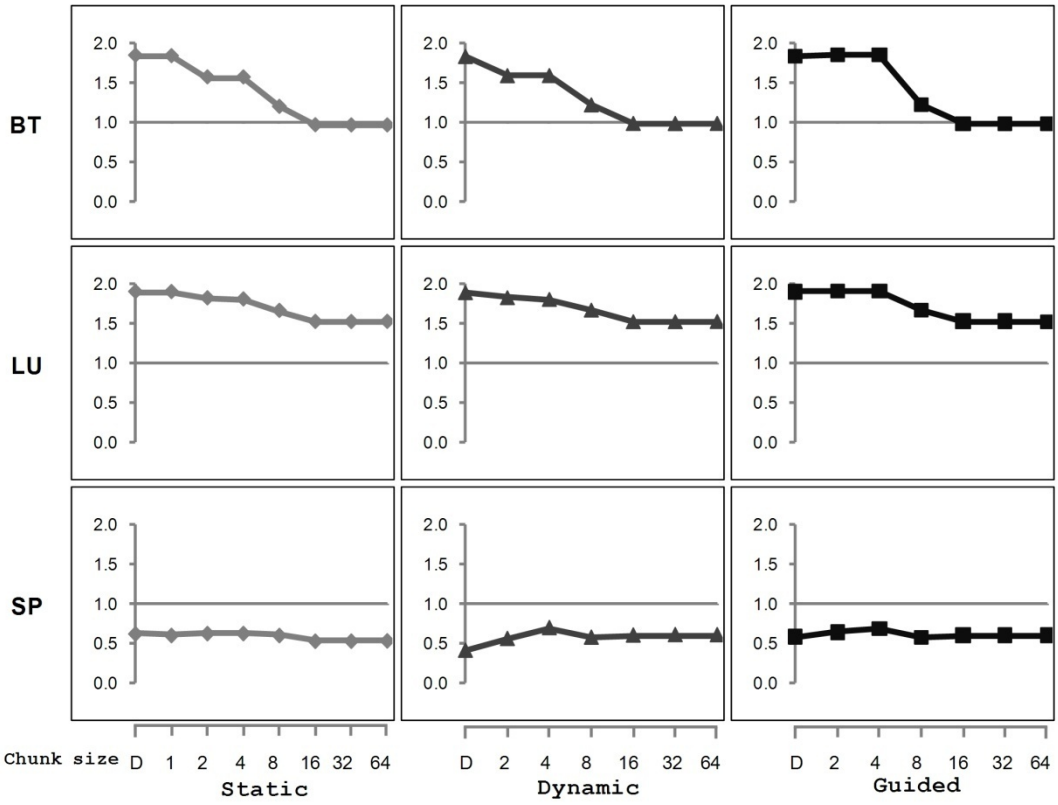
ภาพที่ 5.2 สัดส่วนเวลาประมวลผลของชุดโปรแกรมเอ็นพีบี 2.3



5.1.1 วิธีทำการทดลอง

การทดลองนี้ได้ประมวลผลชุดโปรแกรมเอ็นพีบี 2.3 คลาสเอสด้วยจำนวน 2 เทรด บนชิปมัลติโพรเซสเซอร์ที่มี 2 หน่วยประมวลผลและทุกหน่วยใช้แคชระดับที่ 2 ร่วมกัน โดยมีค่าพารามิเตอร์ดังแสดงในตารางที่ 5.1 การประมวลผลของทั้งแปดโปรแกรมย่อยนั้น ได้ใช้คำสั่ง **static**, **dynamic** และ **guided** ซึ่งแต่ละคำสั่งกำหนดใช้ขนาดของบล็อกโดยปริยาย (Default) และขนาดตั้งแต่ 1 ลูปเพิ่มขึ้นที่ละสองเท่าจนถึง 64 ลูป รวมทั้งหมด 22 คำสั่ง โดยการทดลองนี้เปรียบเทียบสมรรถนะคำสั่งการแจกตัวนับลูป ด้วยค่าสปีดอัพเมื่อเทียบกับการประมวลผลด้วยหนึ่งเทรด

ภาพที่ 5.3 ค่าสปีดอัพของกลุ่มโปรแกรมจำลองการไหลของสารเคมี (บีที แอลยู และเอสพี)

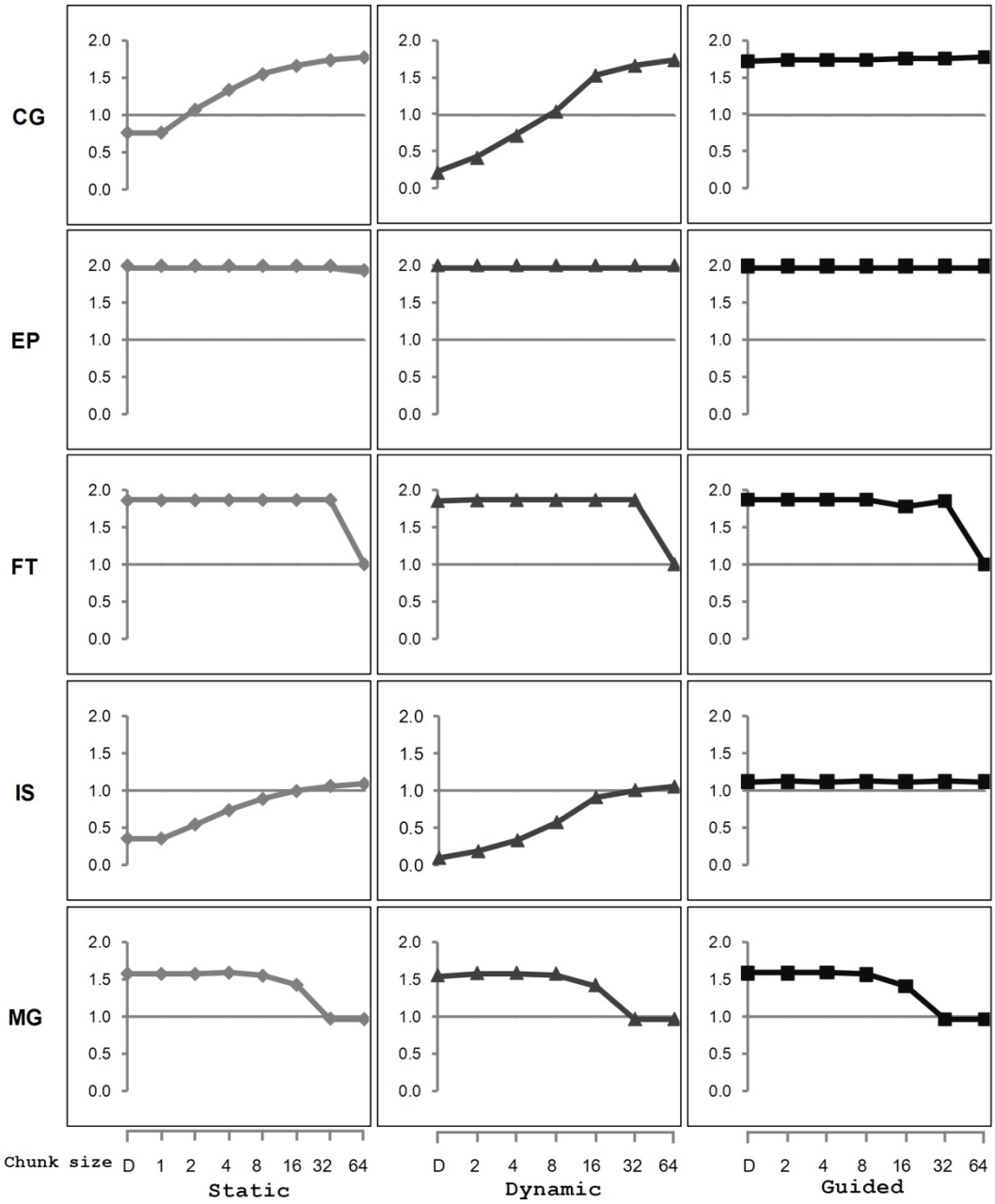


5.1.2 ผลการทดลอง

ผลการทดลองในภาพที่ 5.3 และภาพที่ 5.4 แสดงให้เห็นว่า แต่ละโปรแกรมย่อยมีคำสั่งการแจกตัวนับลูปที่ให้สมรรถนะการประมวลผลสูงที่สุดแตกต่างกัน และหากพิจารณาจาก

แนวโน้มการเปลี่ยนแปลงของค่าสปีดอัพของแต่ละโปรแกรมเมื่อปรับให้บล็อกมีขนาดเพิ่มขึ้น พบว่า มีการเปลี่ยนแปลงสามลักษณะ ประกอบด้วย (1) ค่าสปีดอัพมีแนวโน้มลดลง ได้แก่ โปรแกรมย่อยบีที เอฟที แอลยู และเอ็มจี แต่โปรแกรมย่อยเอฟทีนั้นมีค่าสปีดอัพลดลงเมื่อกำหนดให้บล็อกมีขนาดเท่ากับ 64 ลูป (2) ค่าสปีดอัพมีแนวโน้มเพิ่มขึ้น ได้แก่ โปรแกรมย่อยซีจี และไอเอส เมื่อประมวลผลด้วยคำสั่ง **`static`** และ **`dynamic`** ส่วนการประมวลผลด้วยคำสั่ง **`guided`** มีแนวโน้มเช่นเดียวกับ (3) ค่าสปีดอัพมีแนวโน้มคงที่ ได้แก่ โปรแกรมย่อยอีพีและเอสพี

ภาพที่ 5.4 ค่าสปีดอัพของกลุ่มโปรแกรมภาระงานหลักทางวิทยาศาสตร์ (ซีจี อีพี เอฟที ไอเอส และเอ็มจี)



หากพิจารณาโดยรวมของทั้งแปดโปรแกรมย่อย พบว่า คำสั่ง **guided** ที่กำหนดขนาดของบล็อกโดยปริยายซึ่งเท่ากับ 1 ลูป และปรับเพิ่มขึ้นไม่เกิน 8 ลูป จะมีค่าเฉลี่ยของสปีดอัพสูงที่สุด เนื่องจากคำสั่ง **guided** ใช้วิธีการแจกตัวนับลูปเมื่อโปรแกรมรัน โดยที่คอมไพเลอร์สามารถปรับขนาดของบล็อกที่แจกไปยังแต่ละเทรดในขณะที่โปรแกรมกำลังประมวลผลได้ แต่บล็อกต้องมีขนาดไม่น้อยกว่าที่กำหนดไว้ ด้วยคุณลักษณะดังกล่าวทำให้คำสั่ง **guided** สามารถแจกตัวนับลูปได้ยืดหยุ่นกว่าคำสั่ง **static** และ **dynamic**

5.1.3 สรุปและอภิปรายผลการทดลอง

จากเป้าหมายการทดลองเปรียบเทียบสมรรถนะคำสั่งแจกตัวนับลูป ของชุดโปรแกรมเอ็นพีบี 2.3 ผลการทดลองแสดงให้เห็นว่า (1) ในแต่ละโปรแกรมย่อย มีคำสั่งการแจกตัวนับลูปที่เหมาะสมแตกต่างกันขึ้นอยู่กับคุณลักษณะของโปรแกรม ผลการทดลองได้ยืนยันความสำคัญของการพัฒนาเครื่องมือวิเคราะห์สมรรถนะคำสั่งการแจกตัวนับลูป ของโปรแกรมแบบโอเพนเอ็มพี ซึ่งเป็นวัตถุประสงค์ของโครงงานวิจัยนี้ และ (2) จากการศึกษาวิธีการประมวลผลชุดโปรแกรมเอ็นพีบี 2.3 พบว่า โปรแกรมเอ็นพีบีมีขนาดของข้อมูลไว้ให้ผู้ใช้เลือกประมวลผลห้าคลาส (W, S, A, B และ C เรียงจากน้อยไปหามาก) อย่างไรก็ตาม เมื่อกำหนดข้อมูลให้มีขนาดคลาสเอขึ้นไป การประมวลผลของแต่ละโปรแกรมย่อยบนซิมิกส์ ต้องใช้เวลาประมวลผลมากกว่าสองชั่วโมง ด้วยข้อจำกัดของเวลาจึงเลือกใช้ขนาดข้อมูลคลาสเอสประมวลผลในการทดลองนี้ และจะใช้ประมวลผลเพื่อทดสอบความถูกต้องของพี-สปาต่อไป

5.2 การเปรียบเทียบสมรรถนะของแคชระดับที่ 2 บนซิปมัลติโพรเซสเซอร์

ปัจจุบันซิปมัลติโพรเซสเซอร์ขนาด 2 หน่วยเป็นหน่วยประมวลผลหลักของบริษัทผู้ผลิตชั้นนำ ซึ่งซิปมัลติโพรเซสเซอร์ดังกล่าวมีแนวทางออกแบบแคชระดับที่ 2 ได้สองสถาปัตยกรรม[1] คือ ทุกหน่วยใช้แคชระดับที่ 2 ร่วมกัน (*สถาปัตยกรรมแบบใช้แคชร่วมกัน*) หรือแต่ละหน่วยมีแคชระดับที่ 2 แยกกัน (*สถาปัตยกรรมแบบใช้แคชแยกกัน*) งานวิจัยของเซา (Zhao *et al.*, 2007) ได้เปรียบเทียบสมรรถนะสถาปัตยกรรมของแคชระดับที่ 2 บนซิปมัลติโพรเซสเซอร์ขนาด 4 หน่วย จากผลการทดลอง พบว่า สถาปัตยกรรมแบบใช้แคชร่วมกันมีอัตราการไม่พบ

[1] ตัวอย่างเช่น Intel Core 2 Duo ทุกหน่วยใช้แคชระดับที่ 2 ร่วมกัน ในขณะที่ AMD Athlon X2 แต่ละหน่วยใช้แคชระดับที่ 2 แยกกัน

ข้อมูลในแคชระดับที่ 2 น้อยกว่าแบบใช้แคชแยกกัน การทดลองนี้จึงนำแนวทางดังกล่าวมาทดลองเปรียบเทียบบนชิปมัลติโพรเซสเซอร์ขนาด 2 หน่วย โดยมีเป้าหมายสองประการ คือ (1) เพื่อเปรียบเทียบสมรรถนะของแคชระดับที่ 2 ทั้งสองสถาปัตยกรรม เมื่อประมวลผลชุดโปรแกรมเอ็นพีบี 2.3 และ (2) เพื่อศึกษาการปรับค่าพารามิเตอร์และการเรียกดูข้อมูลการทำงานของแคชบนโมดูลจี-แคช

ตารางที่ 5.1 สถาปัตยกรรมของชิปมัลติโพรเซสเซอร์ที่ใช้เปรียบเทียบสมรรถนะของแคช

สถาปัตยกรรมของชิปมัลติโพรเซสเซอร์	สถาปัตยกรรมแบบใช้แคชร่วมกัน	สถาปัตยกรรมแบบใช้แคชแยกกัน
สถาปัตยกรรมของหน่วยประมวลผล		
- จำนวนหน่วยประมวลผล	2 หน่วย	
- ชุดคำสั่ง	x86_64	
สถาปัตยกรรมของแคชระดับที่ 1		
- ขนาดของแคชระดับที่ 1 (แยกแคชคำสั่งและข้อมูล)	32 KB / private	
- ขนาดของแคชบล็อก	64 bytes	
- จำนวนเซต	8 way	
- นโยบายการเขียน	write through, non-allocate	
- เวลาการเข้าถึง	3 cycles	
สถาปัตยกรรมของแคชระดับที่ 2		
- ขนาดของแคชระดับที่ 2	**2 MB / shared**	**2 x 1 MB / private**
- ขนาดของแคชบล็อก	64 bytes	
- จำนวนเซต	16 way	
- นโยบายการเขียน	write-back, allocate	
- เวลาการเข้าถึง	10 cycles	
- เวลาการเข้าถึงในกรณีไม่พบข้อมูล	165 cycles	
กลไกรักษาความสอดคล้องของข้อมูลในแคช	**ไม่มี**	**MESI แคชระดับที่ 2**
ความจุของหน่วยความจำหลัก	2 GB	

5.2.1 วิธีทำการทดลอง

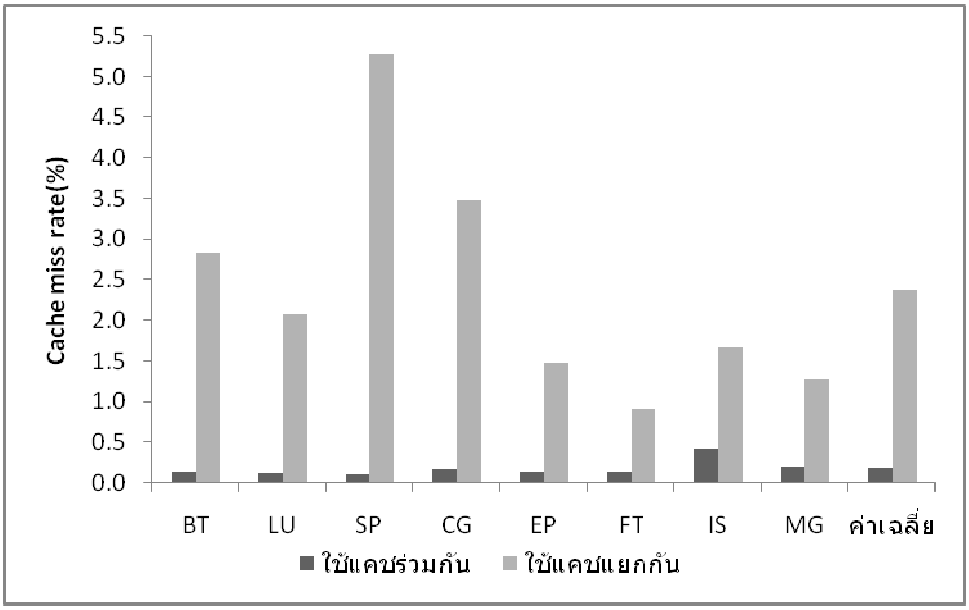
การทดลองนี้ประมวลผลชุดโปรแกรมเอ็นพีบี 2.3 ด้วยจำนวน 2 เทรด บนชิปมัลติโพรเซสเซอร์ที่ใช้แคชระดับที่ 2 ร่วมกันและแยกกัน ตารางที่ 5.1แสดงถึงลักษณะทางสถาปัตยกรรม

ของชิปมัลติโพรเซสเซอร์ทั้งสองสถาปัตยกรรมที่กำหนดไว้บนไฟล์บทคำสั่งของซิมิกส์ ซึ่งค่าพารามิเตอร์ได้อ้างอิงจากหน่วยประมวลผลคอร์ 2 ดูโอ ของบริษัทอินเทล โดยที่แคชระดับที่ 1 แบ่งออกเป็นแคชคำสั่งและแคชข้อมูลแยกออกจากกัน ซึ่งมีขนาด 32 กิโลไบต์ ส่วนแคชระดับที่ 2 มีขนาดรวมเท่ากับ 2 เมกะไบต์ ทั้งแคชระดับที่ 1 และแคชระดับที่ 2 มีขนาดแคชบล็อกเท่ากับ 64 ไบต์เท่ากัน แต่แคชระดับที่ 1 แบ่งแคชออกเป็น 8 เซตสัมพันธ์และใช้เวลาเข้าถึง 3 รอบสัญญาณนาฬิกา ขณะที่แคชระดับที่ 2 จะแบ่งแคชออกเป็น 16 เซตสัมพันธ์และใช้เวลาเข้าถึง 10 รอบสัญญาณนาฬิกา ส่วนเวลาเข้าถึงข้อมูลในหน่วยความจำเท่ากับ 165 รอบสัญญาณนาฬิกา การทดลองนี้ใช้การเปรียบเทียบสมรรถนะของแคชด้วยอัตราการไม่พบข้อมูลในแคชระดับที่ 2

5.2.2 ผลการทดลอง

ผลการทดลอง (ภาพที่ 5.5) แสดงให้เห็นว่า จากการประมวลผลของแปดโปรแกรมย่อยนั้น สถาปัตยกรรมแบบใช้แคชร่วมกันมีอัตราการไม่พบข้อมูลในแคชระดับที่ 2 น้อยกว่าสถาปัตยกรรมแบบใช้แคชแยกกัน และจากการวิเคราะห์สถิติแบบการจำแนกทางเดียว (One-way Anova) พบ ว่า สถาปัตยกรรมแบบใช้แคชร่วมกันมีค่าเฉลี่ยอัตราการไม่พบข้อมูลในแคชระดับที่ 2 น้อยกว่าสถาปัตยกรรมแบบใช้แคชแยกกันร้อยละ 2.19 อย่างมีนัยสำคัญ ที่ระดับความเชื่อมั่นร้อยละ 95 ซึ่งผลการทดลองดังกล่าวสอดคล้องกับงานวิจัยของเซาที่ได้กล่าวถึงข้างต้น

ภาพที่ 5.5 อัตราการไม่พบข้อมูลในแคชระดับที่ 2 ของชุดโปรแกรมเอ็นพีบี 2.3



5.2.3 สรุปและอภิปรายผลการทดลอง

จากเป้าหมายของการทดลองเปรียบเทียบสมรรถนะของแคชระดับที่ 2 บนชิปมัลติโพรเซสเซอร์ ผลการทดลองแสดงให้เห็นว่า (1) สถาปัตยกรรมแบบใช้แคชร่วมกันมีสมรรถนะการทำงานสูงกว่าสถาปัตยกรรมแบบใช้แคชแยกกัน โดยวัดจากอัตราการไม่พบข้อมูลในแคชระดับที่ 2 เนื่องจากหน่วยประมวลผลใช้แคชระดับที่ 2 ร่วมกัน ทำให้แคชมีพื้นที่เพียงพอต่อเก็บข้อมูลทั้งหมดที่โปรแกรมใช้ระหว่างประมวลผล จึงส่งผลให้อัตราการไม่พบข้อมูลในแคชลดลง และจากการศึกษาการทำงานของโมดูลจี-แคช พบว่า (2) จี-แคชมีค่าพารามิเตอร์ที่สามารถปรับใช้ได้อย่างครบถ้วนดังที่แสดงในตารางที่ 5.1 และมีเครื่องมือที่สามารถใช้เรียกดูข้อมูลการทำงานของแคชได้บ้างส่วน ซึ่งโครงงานวิจัยจะพัฒนาให้เครื่องมือสามารถเก็บ รวบรวม และวิเคราะห์ข้อมูลการใช้ประโยชน์พื้นที่แคชได้ละเอียดขึ้น

5.3 การวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูลของการคูณเมทริกซ์แบบขนาน

ฟังก์ชันการคูณเมทริกซ์เป็นหนึ่งในภาระงานหลักของโปรแกรมเพื่อความบันเทิง เกมส์ และการคำนวณทางวิทยาศาสตร์ ดังนั้น การศึกษาคุณสมบัติการใช้พื้นที่เก็บข้อมูลจะช่วยให้เข้าใจพฤติกรรมระหว่างประมวลผลของการคูณเมทริกซ์ได้ละเอียดขึ้น การทดลองนี้จึงมีเป้าหมายสามประการ คือ (1) เพื่อวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูล และเปรียบเทียบค่าสปีดอัพของการคูณเมทริกซ์ เมื่อใช้จำนวนเทรดในการประมวลผลต่างกัน (2) เพื่อศึกษาแนวทางการพัฒนาส่วนการทำงานของพี-สปา ที่ใช้นำเสนอข้อมูลคุณสมบัติของพื้นที่เก็บข้อมูลในรูปแบบของกราฟ ซึ่งเป็นรูปแบบการแสดงผลที่จะใช้ในโครงงานวิจัยนี้

ภาพที่ 5.6 โครงสร้างไฟล์บันทึกการเข้าถึงข้อมูลบนหน่วยความจำของพี-สปา

[No of access] [CPU] [Access types] [Memory address]

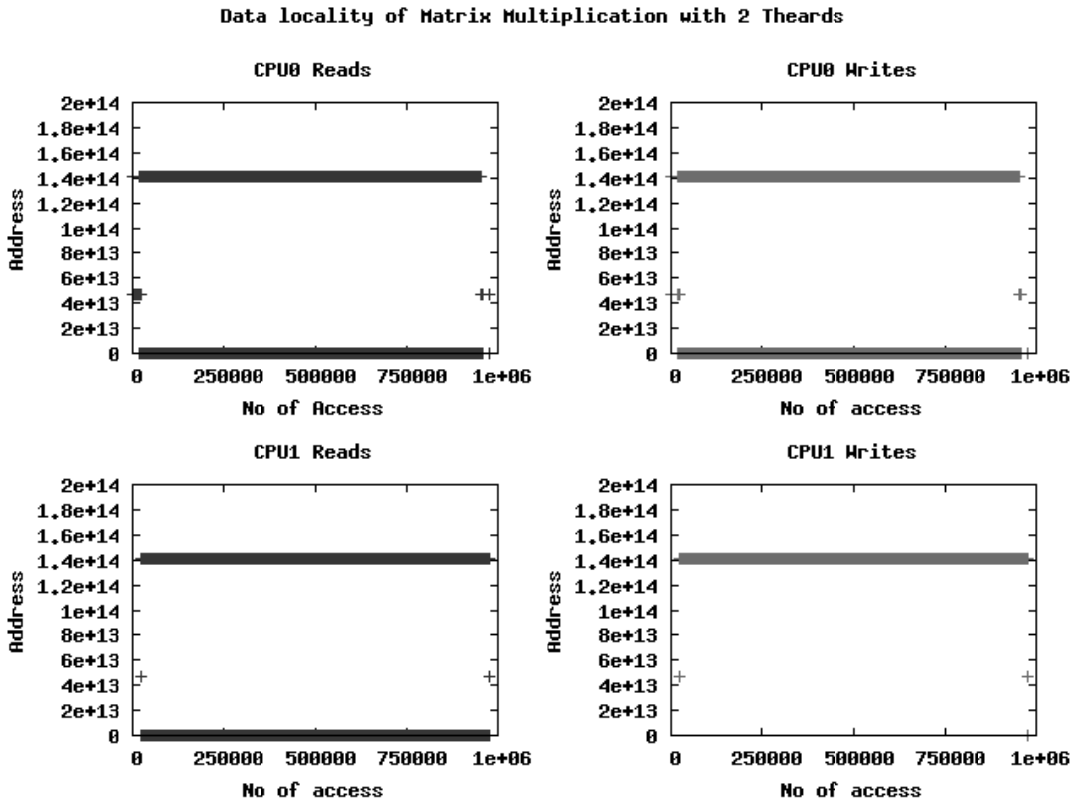
แนวทางการทำงานของส่วนนำเสนอข้อมูลในรูปแบบของกราฟนั้น มีขั้นตอนดังนี้ ในขณะที่โปรแกรมกำลังประมวลผล พี-สปาจะแยกชนิดการเข้าถึงข้อมูลบนหน่วยความจำของแต่ละหน่วยประมวลผล และแยกบันทึกข้อมูลดังกล่าวเป็นไฟล์บันทึกการเข้าถึงข้อมูล โดยไฟล์ดังกล่าวมีโครงสร้างการบันทึกข้อมูลดังที่แสดงภาพที่ 5.6 ซึ่งประกอบด้วย ลำดับการเข้าถึงข้อมูล หน่วยประมวลผล ชนิดการเข้าถึง และตำแหน่งพื้นที่เก็บข้อมูลบนหน่วยความจำ เมื่อโปรแกรม

ประมวลผลเสร็จเรียบร้อย ไฟล์ดังกล่าวจะถูกส่งต่อไปยังพี-กราฟเพื่อใช้เป็นข้อมูลให้โปรแกรมกนูพล็อตนำไปสร้างกราฟ ซึ่งรูปแบบของกราฟนั้นจะถูกกำหนดด้วยไฟล์บทคำสั่งที่อยู่ในพี-กราฟ

5.3.1 วิธีทำการทดลอง

การทดลองนี้ประมวลผลฟังก์ชันการคูณเมทริกซ์ ด้วยเทรดจำนวน 2, 4, 8, 16 และ 32 เทรด บนชิปมัลติโพรเซสเซอร์ที่มี 2 หน่วยประมวลผลและใช้แคชระดับที่ 2 ร่วมกัน โดยมีค่าพารามิเตอร์ดังแสดงไว้ในตารางที่ 5.1 ขนาดของเมทริกซ์ที่ใช้ในการประมวลผลเท่ากับครึ่งหนึ่งของพื้นที่แคชระดับที่ 1 (อาร์เรย์ของเลขจำนวนจริงขนาด 32x32 ช่อง) เพื่อให้สามารถดึงข้อมูลทั้งหมดเข้ามาประมวลผลบนแคชระดับที่ 1 ได้พร้อมกัน การทดลองนี้จะเปรียบเทียบคุณสมบัติของพื้นที่เก็บข้อมูล และค่าสปีดอัพของโปรแกรม เมื่อประมวลผลด้วยจำนวนเทรดต่างกัน

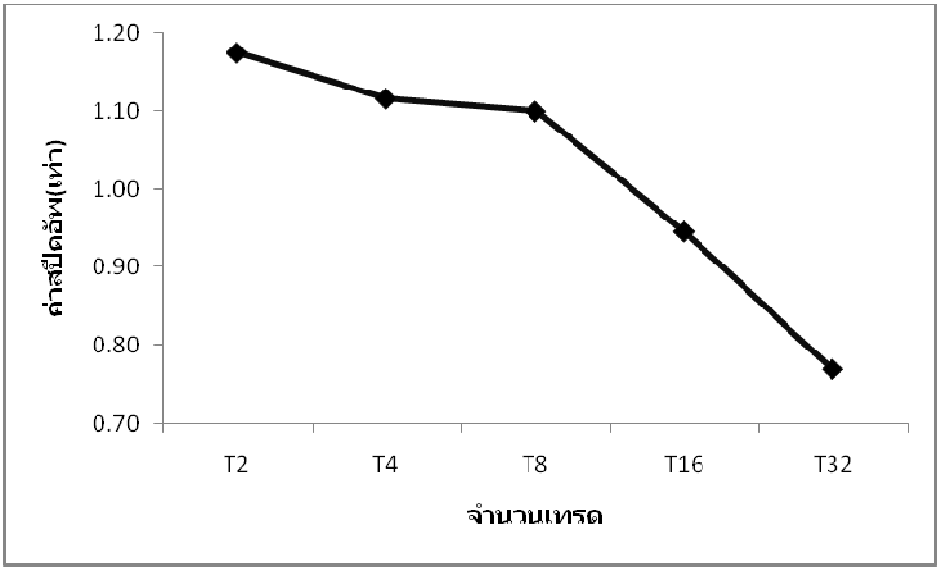
ภาพที่ 5.7 คุณสมบัติการใช้พื้นที่เก็บข้อมูลของฟังก์ชันการคูณเมทริกซ์ บน 2 เทรด



5.3.2 ผลการทดลอง

ผลการวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูล (ภาพที่ 5.7) พบว่า การประมวลผลฟังก์ชันการคูณเมทริกซ์ ด้วยจำนวน 2, 4, 8, 16 และ 32 เทรด มีคุณสมบัติของพื้นที่เก็บข้อมูลในลักษณะเดียวกัน คือ พื้นที่ที่ถูกใช้บ่อยครั้ง จากภาพที่ 5.7 แสดงการเข้าถึงข้อมูลของทั้งสองหน่วยประมวลผล เมื่อประมวลผลด้วย 2 เทรด ซึ่งจะเห็นได้ว่า การอ่านและเขียนข้อมูลของทั้งสองหน่วยประมวลผลเป็นการเข้าถึงข้อมูลในพื้นที่เดิมบ่อยครั้ง แต่ทว่าทั้งสองหน่วยประมวลผลเข้าถึงข้อมูลบนพื้นที่เดียวกัน ซึ่งอาจทำให้ต้องใช้ข้อมูลร่วมกันบ่อยครั้งและทำให้เกิดอัตราการไม่พบข้อมูลเนื่องจากใช้ข้อมูลร่วมกันสูงขึ้นด้วย

ภาพที่ 5.8 ค่าสปีดอัพของฟังก์ชันการคูณเมทริกซ์ บน 2, 4, 8, 16 และ 32 เทรด



แม้ว่าการประมวลผลฟังก์ชันการคูณเมทริกซ์บนทุกเทรดให้คุณสมบัติการใช้พื้นที่เก็บข้อมูลที่เหมือนกัน แต่จากภาพที่ 5.8 แสดงให้เห็นว่า ค่าสปีดอัพของโปรแกรมลดลงเมื่อปรับจำนวนเทรดที่ใช้ประมวลผลเพิ่มขึ้น เนื่องจากจำนวนเทรดที่เพิ่มขึ้นส่งผลให้โปรแกรมต้องใช้เวลาส่วนเกินเพื่อประสานจังหวะการทำงานระหว่างเทรดสูงขึ้น ซึ่งทำให้โปรแกมสูญเสียสมรรถนะการประมวลผล จากผลการทดลองจะเห็นได้ว่า จำนวนเทรดมีผลต่อสมรรถนะการประมวลผลของฟังก์ชันการคูณเมทริกซ์เช่นกัน

5.3.3 สรุปและอภิปรายผลการทดลอง

จากเป้าหมายของการทดลองเพื่อวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูลของฟังก์ชันการคูณเมทริกซ์แบบขนาน ผลการทดลองแสดงให้เห็นว่า (1) การคูณเมทริกซ์มีคุณสมบัติการใช้พื้นที่เก็บข้อมูลเดิมบ่อยครั้ง อย่างไรก็ตาม ทุกหน่วยประมวลผลจะเข้าถึงพื้นที่เก็บข้อมูลตำแหน่งเดียวกัน ทำให้มีโอกาสเกิดการไม่พบข้อมูลในแคชเนื่องจากการใช้ข้อมูลร่วมกันในอัตราที่สูง ซึ่งการจำแนกชนิดของการไม่พบข้อมูล จะช่วยแจกแจงโอกาสที่จะเกิดการไม่พบข้อมูลด้วยสาเหตุดังกล่าวได้ และทำให้เข้าใจข้อจำกัดของโปรแกรมได้ดียิ่งขึ้น โครงงานวิจัยนี้จึงมีเป้าหมายที่จะพัฒนาให้พี-สปาสามารถจำแนกชนิดของการไม่พบข้อมูลได้ นอกจากนี้ จากการเปรียบเทียบค่าสปีดอัพยังพบว่า เมื่อจำนวนเทรดที่ใช้ประมวลผลเพิ่มขึ้นจะทำให้สมรรถนะการประมวลผลของฟังก์ชันการคูณเมทริกซ์ลดลง

จากการศึกษาแนวทางการนำเสนอข้อมูลของพี-สปา พบว่า (2) แม้ว่าพี-สปาจะสามารถสร้างไฟล์บันทึกการเข้าถึงข้อมูลเพื่อนำไปแสดงผลในรูปแบบของกราฟได้ แต่โครงงานวิจัยนี้ยังประสบปัญหาเนื่องจากไฟล์บันทึกการเข้าถึงข้อมูลมีขนาดใหญ่มากเกินไป ซึ่งแนวทางการแก้ปัญหาดังกล่าวจะต้องลดรายละเอียดและความถี่ของการบันทึกข้อมูลของพี-สปาให้น้อยลง นอกจากนี้ การนำเสนอคุณสมบัติการใช้พื้นที่เก็บข้อมูลในรูปแบบของกราฟ เป็นอีกแนวทางหนึ่งที่จะช่วยให้ผู้ใช้เข้าพฤติกรรมของโปรแกรมระหว่างการประมวลผล และทำให้ทราบข้อจำกัดเชิงสมรรถนะของโปรแกรมได้ดีขึ้น

5.4 สรุปผลการทดลองเบื้องต้น

การทดลองเบื้องต้นที่กล่าวถึงในบทนี้ เป็นการทดลองเพื่อศึกษาวิธีการทำงานและการเก็บข้อมูลของระบบจำลองซิมิกส์ เพื่อนำไปใช้เป็นแนวการพัฒนาพี-สปาให้สามารถทำงานร่วมกับซิมิกส์ได้อย่างถูกต้อง ซึ่งได้แบ่งการทดลองออกเป็นสามงานทดลอง ตามส่วนการทำงานของพี-สปา ได้แก่ *ส่วนที่หนึ่ง* ศึกษาการประมวลผลโปรแกรมแบบโอเพนเอ็มพีบนซิมิกส์ ซึ่งได้ทดลองเพื่อเปรียบเทียบสมรรถนะคำสั่งการแจกตัวนับลูปของชุดโปรแกรมเอ็นพีบี 2.3 *ส่วนที่สอง* ศึกษาการปรับค่าพารามิเตอร์และการเรียกดูข้อมูลจากโมดูลจี-แคช โดยได้ทดลองเปรียบเทียบสมรรถนะของแคชระดับที่ 2 บนชิปมัลติโพรเซสเซอร์ และ*ส่วนที่สาม* เพื่อเป็นแนวทางการพัฒนาและศึกษาส่วนการทำงานของพี-สปา ที่ใช้นำเสนอคุณสมบัติการใช้พื้นที่เก็บข้อมูลในรูปแบบของกราฟ ซึ่งในส่วนนี้ได้ทดลองเพื่อวิเคราะห์คุณสมบัติดังกล่าวบนฟังก์ชันการคูณเมทริกซ์แบบขนาน

การทดลองเบื้องต้นที่ได้กล่าวถึงในบทนี้ ได้ยืนยันว่าแนวทางการดำเนินการวิจัยที่ได้ออกแบบไว้สามารถดำเนินการได้จริงตามกรอบเวลาที่กำหนดไว้ ซึ่งในบทถัดไปเป็นนำเครื่องมือพี-สปาที่พัฒนาเสร็จสิ้นแล้ว ไปวิเคราะห์และแสดงแนวทางการปรับสมรรถนะทั้งในเชิงการออกแบบทางสถาปัตยกรรมของแคชบนชิปมัลติโพรเซอร์ และการเลือกใช้คำสั่งแจกตัวนับลูปที่เหมาะสมกับคุณลักษณะของโปรแกรม เพื่อให้โปรแกรมมีสมรรถนะที่สูงขึ้น