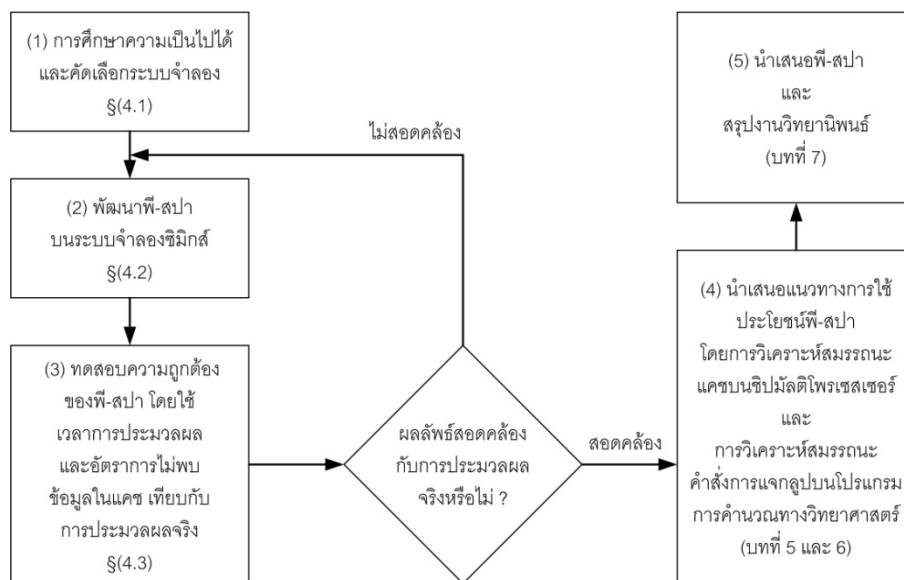


บทที่ 4

วิธีการวิจัย

วิทยานิพนธ์เพื่อพัฒนาเครื่องมือวิเคราะห์สมรรถนะพี-สพานี้ ได้นำระบบจำลองมาใช้เป็นแกนกลางในการพัฒนา เพื่อให้สามารถดำเนินงานวิจัยบรรลุวัตถุประสงค์ดังที่กำหนดไว้ในบทที่ 1 ในบทนี้กล่าวถึงแผนการดำเนินงานวิจัย ซึ่งแบ่งออกเป็นสี่ขั้นตอน (ภาพที่ 4.1) ได้แก่ **ขั้นตอนที่หนึ่ง** เป็นการศึกษาความเป็นไปได้เพื่อคัดเลือกระบบจำลอง ศึกษาแนวทางการเก็บข้อมูล และกำหนดสภาพแวดล้อมในการทำวิจัย **ขั้นตอนที่สอง** เป็นการศึกษาเพื่อพัฒนาเครื่องมือวิเคราะห์สมรรถนะพี-สพา **ขั้นตอนที่สาม** เป็นการทดสอบความถูกต้องของพี-สพา **ขั้นตอนที่สี่** เป็นการทำการทดลองเพื่อแสดงศักยภาพของพี-สพาในการวิเคราะห์สมรรถนะของโปรแกรมแบบโอเพนเอ็มพี และ **ขั้นตอนสุดท้าย** เป็นการนำเสนอเครื่องมือพี-สพาและผลงานวิทยานิพนธ์ขึ้นนี้

ภาพที่ 4.1 ภาพรวมของวิธีการดำเนินงานวิจัยของการโครงการพัฒนาเครื่องมือพี-สพา



4.1 การศึกษาความเป็นไปได้

ความท้าทายของเครื่องมือวิเคราะห์สมรรถนะระบบคอมพิวเตอร์ คือการพยายามให้เครื่องมือวิเคราะห์มีโครงสร้างที่สามารถเปลี่ยนแปลงตามสถาปัตยกรรมคอมพิวเตอร์ได้ จากการ

ทบทวนงานวิจัยที่ผ่านมาพบว่าการสร้างเครื่องมือวิเคราะห์บนระบบจำลองที่ผ่านการตรวจสอบความถูกต้อง และได้รับความนิยมทั้งในวงการศึกษและจากบริษัทผู้ผลิตคอมพิวเตอร์นั้น จะเป็นแนวทางให้เครื่องมือวิเคราะห์สามารถรับเข้ากับสถาปัตยกรรมใหม่ได้ โดยไม่เสียเวลาเขียนโปรแกรมเพื่อพัฒนาโมเดลที่เป็นตัวแทนสถาปัตยกรรมนั้นขึ้นมาใหม่ทั้งหมด ดังนั้น งานวิจัยนี้จึงวางแผนงานวิจัยเพื่อพัฒนาเครื่องมือวิเคราะห์บนระบบจำลอง โดยขั้นตอนแรกของงานวิจัยเป็นการคัดเลือกกระบบจำลองที่ถูกนำมาใช้มากที่สุดโดยการอ้างอิงจากงานวิจัยและจากบริษัทผู้ผลิตเพื่อนำระบบจำลองที่คัดเลือกได้ มาเป็นแกนกลางสำหรับพัฒนาส่วนวิเคราะห์การใช้ประโยชน์พื้นที่แคชเพิ่มเติมเข้าไป ทั้งนี้จุดสำคัญในการคัดเลือกอยู่ที่ว่าระบบจำลองที่คัดเลือกขึ้นมานั้นจะต้องรองรับการประมวลผลโปรแกรมโอเพนเอ็มพีบนชิปัลติโปรเซสเซอร์ได้

ตารางที่ 4.1 คุณลักษณะของระบบจำลองสองระบบที่นำมาศึกษาในวิทยานิพนธ์นี้

คุณลักษณะ	เฮสย์	ซิมิกส์
1. เทคนิคของระบบจำลอง	Trace-driven	Full-system
2. โปรแกรมนำเข้า	ภาษาแอสเซมบลี	ภาษาระดับสูง
3. ระบบปฏิบัติการของคอมพิวเตอร์แม่ข่าย	Linux	Unix/Linux
	Cygwin/Windows	Solaris, Windows
4. ระบบปฏิบัติการของคอมพิวเตอร์เป้าหมาย	-	Unix/Linux
		Solaris, Windows
5. สถาปัตยกรรมของหน่วยประมวลผลที่มีในระบบจำลอง	x86	x86/x86_64/IA-64
	SPARC	SPARC/PowerPC

4.1.1 การคัดเลือกระบบจำลอง

ผลการคัดเลือกระบบจำลองจากการทบทวนงานวิจัยที่ผ่านมาพบว่า ระบบจำลองสองระบบ คือ ระบบจำลองเฮสย์และระบบจำลองซิมิกส์ (ตารางที่ 4.1) ซึ่งมีคุณลักษณะสอดคล้องกับเป้าหมายที่ตั้งไว้

ระบบจำลองเฮสย์ (Coe *et al.*, 1998) เป็นระบบจำลองชนิดเหตุการณ์ไม่ต่อเนื่อง ที่ถูกพัฒนาขึ้นมาเพื่อใช้ศึกษาการทำงานของสถาปัตยกรรมคอมพิวเตอร์ โดยได้จัดเตรียมสถาปัตยกรรมพร้อมใช้ส่วนหนึ่ง และมีเครื่องมือเพื่อกำหนดโมเดลสถาปัตยกรรมคอมพิวเตอร์ขึ้นมาใหม่ได้ เฮสย์มีกลไกขับเคลื่อนโมเดลจำลองสถาปัตยกรรมคอมพิวเตอร์ มีการสร้างไฟล์บันทึกขั้นตอนการทำงานระหว่างการผลิตผล และจุดเด่นของระบบจำลองเฮสย์คือสามารถนำ

ไฟล์บันทึกดังกล่าวมาแสดงเป็นภาพเคลื่อนไหวได้ ซึ่งภาพเคลื่อนไหวนี้อำนวยความสะดวกให้ผู้ใช้งานสามารถตรวจสอบความถูกต้องโดยใช้ควบคุมการแสดงผลภาพเคลื่อนไหวเพื่อดูลำดับการทำงาน ทำให้ผู้ใช้เข้าใจขั้นตอนการทำงานของสถาปัตยกรรมคอมพิวเตอร์ได้เร็วขึ้น

ระบบจำลองซิมิกส์ (Magnusson *et al.*, 2002) เป็นระบบจำลองแบบเต็มระบบ (Full-system simulation) ที่สามารถจำลองการทำงานของระบบคอมพิวเตอร์ได้ตั้งแต่กระบวนการบูตอัพเครื่องและโหลดระบบปฏิบัติการ ซิมิกส์สามารถสร้างระบบคอมพิวเตอร์เป้าหมายที่ถูกควบคุมด้วยระบบปฏิบัติการที่แตกต่างกันได้สี่ระบบ (ตารางที่ 4.1) จึงเหมือนกับซิมิกส์สร้างระบบคอมพิวเตอร์เสมือนที่มีแพลตฟอร์มเป็น วินโดวส์ ยูนิกซ์ ลินุกซ์ หรือโซลาริส และประมวลผลแอปพลิเคชันบนแพลตฟอร์มเสมือนนี้ได้ตามที่ใช้กำหนด ซึ่งจุดนี้เป็นข้อได้เปรียบที่ทำให้มีงานวิจัยที่เลือกศึกษาสมรรถนะของคอมพิวเตอร์บนซิมิกส์เป็นจำนวนมาก

4.1.1.1 การทดลองเพื่อศึกษาศักยภาพของระบบจำลองเฮสย์

เพื่อศึกษาศักยภาพของระบบจำลองเฮสย์ ในขั้นแรกผู้วิจัย (พีระศักดิ์ วังโสม และ วรวรรณ มะเรงสิทธิ์, 2550) ได้นำระบบจำลองเฮสย์มาพัฒนาเพิ่มเติมให้เป็นแพลตฟอร์ม เพื่อทดลองเปรียบเทียบสมรรถนะอัลกอริทึมการคูณเมทริกซ์แบบขนานบนชิปมัลติโพรเซสเซอร์ ซึ่งผลการทดลองเพื่อศึกษาศักยภาพของระบบจำลองเฮสย์พบว่า แม้ว่าจุดเด่นของระบบเฮสย์คือการแสดงผลภาพเคลื่อนไหวที่สามารถแสดงขั้นตอนการทำงานของระบบคอมพิวเตอร์อย่างละเอียด แต่ทว่าเฮสย์จะรับโปรแกรมที่ใช้ประมวลผลในรูปแบบของภาษาแอสเซมบลีตามที่กำหนดไว้เท่านั้น ทำให้ก่อนศึกษาสมรรถนะของโปรแกรม ผู้ใช้จะต้องแปลงโปรแกรมจากภาษาระดับสูงให้เป็นภาษาแอสเซมบลีก่อน ซึ่งจุดนี้จะจำกัดมิให้เครื่องมือวิเคราะห์สมรรถนะก้าวตามทันสถาปัตยกรรมเครื่องที่มีชุดคำสั่งใหม่ หรือมีภาษาแอสเซมบลีรูปแบบใหม่ และมีความยุ่งยากในการรับอินพุตที่เป็นโปรแกรมแบบโอเพนเอ็มพีโดยตรง

4.1.1.2 การทดลองเพื่อศึกษาศักยภาพของระบบจำลองซิมิกส์

เพื่อศึกษาศักยภาพของระบบจำลองซิมิกส์ ระบบจำลองซิมิกส์ได้ถูกนำมาปรับเป็นแพลตฟอร์มสำหรับทำการทดลองวิเคราะห์สมรรถนะการทำงานของแคช โดยการทดลองนี้เป็นการทดลองซ้ำกับงานวิจัยของฟาโรกี (Faroughi, 2005) ซึ่งภายหลังจากดำเนินการทดลองซ้ำพบข้อได้เปรียบสองประการของระบบซิมิกส์เมื่อเทียบกับระบบเฮสย์ คือ

1. ซิมิกส์มีโมดูลจี-แคช (G-cache) ที่ทำหน้าที่เสมือนแคช ซึ่งผู้ใช้สามารถพัฒนาโมดูลดังกล่าวเพิ่มเติมเพื่อให้ทำงานตามที่ต้องการได้

2. ระบบคอมพิวเตอร์เป้าหมายบนซิมิกส์สามารถประมวลผลโปรแกรมแบบโอเพนเอ็มพีได้โดยตรงบนทุกคอมพิวเตอร์ และมีชิปมัลติโพรเซสเซอร์ของบริษัทผู้ผลิตชั้นนำอย่างน้อยสี่บริษัท¹ เพื่อให้ผู้ใช้เลือกนำไปศึกษาและวิจัยได้

ด้วยเหตุนี้ ซิมิกส์จึงถูกคัดเลือกเป็นระบบจำลองต้นแบบ เพื่อนำไปพัฒนาเครื่องมือวิเคราะห์สมรรถนะสำหรับโปรแกรมแบบโอเพนเอ็มพี บนชิปมัลติโพรเซสเซอร์ตามวัตถุประสงค์ที่กำหนดไว้ ซึ่งในหัวข้อถัดไปจะกล่าวถึง ส่วนการทำงานหลักและการเก็บข้อมูลของซิมิกส์ที่เกี่ยวกับการวิเคราะห์การใช้ประโยชน์พื้นที่แคช

4.1.2 การเก็บข้อมูลการใช้ประโยชน์พื้นที่แคชจากซิมิกส์

ภายหลังจากคัดเลือกระบบจำลองซิมิกส์เพื่อใช้เป็นแกนกลางในการสร้างแพลตฟอร์มของเครื่องคอมพิวเตอร์เป้าหมาย สำหรับพัฒนาเครื่องมือวิเคราะห์พี-สเปาเพิ่มเติมเข้าไปนั้น ขั้นตอนถัดมาเป็นการศึกษาส่วนการทำงานของซิมิกส์ที่ใช้สร้างระบบคอมพิวเตอร์เป้าหมาย และการเก็บข้อมูลพฤติกรรมของโปรแกรมระหว่างประมวลผล รวมถึงการศึกษาและกำหนดเครื่องมือที่ใช้ในการพัฒนาเครื่องมือวิเคราะห์พี-สเปาบนซิมิกส์ ซึ่งมีรายละเอียดดังนี้

4.1.2.1 การสร้างระบบคอมพิวเตอร์เป้าหมายบนซิมิกส์

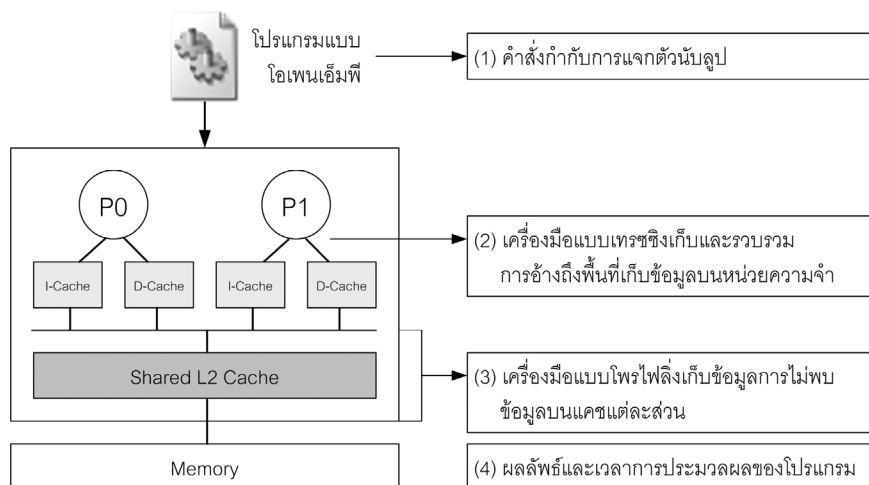
ซิมิกส์เป็นระบบจำลองแบบเต็มระบบที่มีฮาร์ดแวร์เสมือน เพื่อให้ผู้ใช้ปรับแต่งค่าพารามิเตอร์ให้ตรงกับระบบคอมพิวเตอร์เป้าหมายได้ เช่น สถาปัตยกรรมและจำนวนของหน่วยประมวลผล สถาปัตยกรรมและลำดับชั้นของแคช หรือขนาดหน่วยความจำหลักและดิสก์ ค่าพารามิเตอร์เหล่านี้จะต้องถูกกำหนดลงบนไฟล์บทคำสั่ง (Script file) เพื่อให้ซิมิกส์เริ่มทำงานจะเข้าไปอ่านค่าพารามิเตอร์บทไฟล์บทคำสั่งที่ผู้ใช้ระบุไว้ก่อน เพื่อเตรียมสร้างระบบคอมพิวเตอร์เป้าหมายให้ตรงกับความต้องการของผู้ใช้ ซึ่งวิธีการเขียนไฟล์บทคำสั่งเพื่อสร้างระบบคอมพิวเตอร์เป้าหมายสามารถศึกษาได้จากคู่มือการใช้งานซิมิกส์ (Simics, 2007)

¹ อินเทล เอเอ็มดี ซัน และไอบีเอ็ม

4.1.2.2 การเก็บข้อมูลพฤติกรรมการประมวลผลจากซิมิกส์

บนระบบจำลองซิมิกส์มีเครื่องมือทั้งแบบเทรซซิงและโพรไฟลิง ที่ผู้ใช้สามารถนำไปใช้เก็บและรวบรวมข้อมูลบนฮาร์ดแวร์ได้ รวมถึงการเก็บข้อมูลการใช้ประโยชน์พื้นที่แคชด้วย จากภาพที่ 4.2 แสดงวิธีการเก็บข้อมูลการใช้ประโยชน์พื้นที่แคชของซิมิกส์ ที่เกิดขึ้นระหว่างการประมวลผลโปรแกรมแบบโอเพนเอ็มพีบนชิปมัลติโพรเซสเซอร์ โดยเริ่มจาก (1) ระบบคอมพิวเตอร์เป้าหมายบนซิมิกส์ประมวลผลโปรแกรมแบบโอเพนเอ็มพี เมื่อโปรแกรมประมวลผลมาถึงคำสั่งกำกับการแจกตัวนับรูปหรือส่วนของโปรแกรมที่ต้องการเก็บข้อมูลเชิงสมรรถนะ (2) เครื่องมือแบบเทรซซิงก็จะเก็บและรวบรวมการอ้างถึงข้อมูลบนหน่วยความจำ (Memory reference) ของหน่วยประมวลผลที่เกิดขึ้นทั้งหมด ในขณะที่ (3) เครื่องมือแบบโพรไฟลิงจะเก็บข้อมูลการใช้ประโยชน์พื้นที่แคชเบื้องต้น ซึ่งจะเก็บข้อมูลจากโมดูลจี-แคชที่ทำหน้าที่เป็นแคชเสมือน ข้อมูลที่มีได้แก่จำนวนการเข้าถึงข้อมูลทั้งหมด และจำนวนการไม่พบข้อมูลบนแคชในแต่ละลำดับชั้น และ (4) เมื่อโปรแกรมประมวลผลสิ้นสุด ซิมิกส์จะแสดงผลลัพธ์ของโปรแกรมและเวลาประมวลผลของโปรแกรมแก่ผู้ใช้

ภาพที่ 4.2 วิธีการเก็บข้อมูลการใช้ประโยชน์พื้นที่แคชของซิมิกส์ บนชิปมัลติโพรเซสเซอร์ 2 หน่วยประมวลผล



4.1.2.3 เครื่องมือและสภาพแวดล้อมการพัฒนา

งานวิจัยนี้ได้พัฒนาพี-สแปนโปรแกรมระบบจำลองซิมิกส์ เวอร์ชัน 3.0.31 และใช้โปรแกรมกนูพล็อต เวอร์ชัน 4.2 เพื่อให้นำเสนอข้อมูลในรูปแบบของกราฟ โดยทั้งสองโปรแกรม

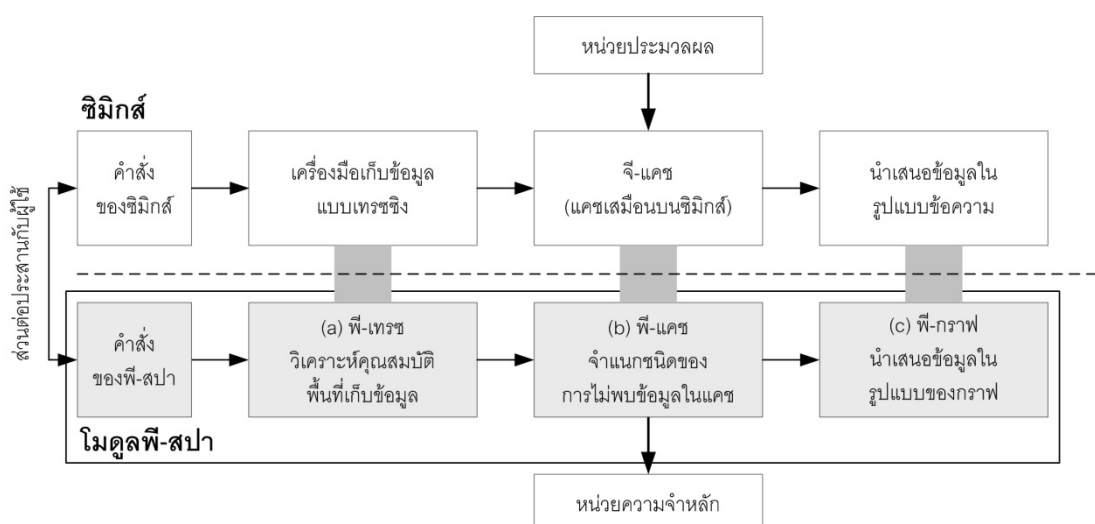
ทำงานบนระบบปฏิบัติการลินุกซ์ อูบันตุ (Ubuntu) เวอร์ชัน 8.0.4 ซึ่งติดตั้งบนเครื่องคอมพิวเตอร์ส่วนบุคคล ที่ใช้หน่วยประมวลผลแบบชิปัลติโพรเซสเซอร์ 2 หน่วย ยี่ห้อเอเอ็มดีที่มีความเร็วสัญญาณนาฬิกา 2.10 กิกะเฮิร์ตซ์ และมีหน่วยความจำหลักขนาด 2 กิกะไบต์

แม้ว่าซิมิกส์จะมีส่วนการทำงานที่สามารถใช้วิเคราะห์การใช้ประโยชน์พื้นที่แคชได้ แต่ทว่าข้อมูลที่ได้จากการวิเคราะห์ยังไม่ครบถ้วนตามวัตถุประสงค์ของงานวิจัยนี้ ขั้นตอนถัดไปจึงเป็นการพัฒนาเครื่องมือวิเคราะห์พี-สปา เพื่อให้สามารถนำเสนอข้อมูลได้ตามเป้าหมาย ซึ่งจะกล่าวถึงรายละเอียดการพัฒนาในหัวข้อถัดไป

4.2 การพัฒนาเครื่องมือวิเคราะห์พี-สปา

การพัฒนาเครื่องมือวิเคราะห์พี-สปาเพื่อใช้เป็นเครื่องมือวิเคราะห์การใช้ประโยชน์พื้นที่แคชบนโปรแกรมแบบโอเพนเอ็มพี ดังที่กำหนดเป็นวัตถุประสงค์ของงานวิจัยนั้น โมดูลพี-สปา ได้ถูกพัฒนาเป็นเพิ่มเติมบนซิมิกส์ เพื่อให้พี-สปาสามารถทำงานได้บนทุกแพลตฟอร์มที่ซิมิกส์รองรับ ซึ่งจะช่วยให้ผู้ใช้เปรียบเทียบการทำงานข้ามแพลตฟอร์มได้ง่ายขึ้น การทำงานของพี-สปาแบ่งออกเป็นสามส่วน (ภาพที่ 4.3) ประกอบด้วย พี-เทรซ (P-trace) เป็นส่วนการวิเคราะห์คุณสมบัติการเข้าถึงพื้นที่ข้อมูล พี-แคช (P-cache) เป็นส่วนการจำแนกชนิดการไม่พบข้อมูลในแคช และ พี-กราฟ (P-graph) เป็นส่วนการนำเสนอข้อมูลในรูปแบบของกราฟ นอกจากนี้จะกล่าวสรุปส่วนการทำงานและความสามารถของพี-สปาที่เพิ่มเติมจากซิมิกส์ไว้ในส่วนท้าย ซึ่งการทำงานของแต่ละส่วนมีรายละเอียดดังนี้

ภาพที่ 4.3 โมดูลพี-สปามีส่วนการทำงานสามส่วนประกอบด้วย (a) พี-เทรซ (b) พี-แคช และ (c) พี-กราฟ



4.2.1 ส่วนการวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูล

ส่วนการทำงานแรกคือ โมดูลพี-เทรซเป็นส่วนการวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูล พี-เทรซจะทำงานร่วมกับเครื่องมือแบบเทรซซิงของซิมิกส์ ซึ่งทำหน้าที่เก็บลำดับและรายละเอียดการเข้าถึงข้อมูลของหน่วยประมวลผลที่ส่งไปยังแคช ส่วนการทำงานของพี-เทรซถูกแทรกเข้าไปในเครื่องมือเทรซซิงเพื่อดึงรายละเอียดดังกล่าวนำมาวิเคราะห์ ผู้ใช้สามารถกำหนดขนาดของพื้นที่ที่ต้องการวิเคราะห์ได้ตั้งแต่ วิเคราะห์ระดับเซกเมนต์ (Segment) ระดับเพจ (Page) ระดับแคชบล็อก และระดับไบต์ โดยพี-เทรซจะแสดงผลการวิเคราะห์ข้อมูลตามขนาดที่ผู้ใช้กำหนด ขั้นตอนการวิเคราะห์และการเก็บข้อมูลออกจากเป็นสองขั้นตอน คือ การจำแนกรูปแบบพื้นที่หน่วยความจำ และการบันทึกข้อมูลลงบนเทรซไฟล์

ภาพที่ 4.4 ตัวอย่างการจำแนกรูปแบบพื้นที่หน่วยความจำแปดรูปแบบของพี-เทรซ

Cycle 1 P0 Read Cycle 2 P0 Read Cycle 3 P0 Read Cycle 4 P0 Read Cycle 5 P0 Read	Cycle 1 P0 Write Cycle 2 P0 Write Cycle 3 P0 Write Cycle 4 P0 Write Cycle 5 P0 Write	Cycle 1 P0 Read Cycle 2 P1 Read Cycle 3 P0 Read Cycle 4 P0 Read Cycle 5 P1 Read	Cycle 1 P0 Write Cycle 2 P1 Write Cycle 3 P1 Write Cycle 4 P0 Write Cycle 5 P0 Write
(1) SR	(2) SW	(3) MR	(4) MW
Cycle 1 P0 Read Cycle 2 P0 Read Cycle 3 P0 Write Cycle 4 P0 Read Cycle 5 P0 Write	Cycle 1 P0 Read Cycle 2 P0 Read Cycle 3 P1 Write Cycle 4 P0 Read Cycle 5 P0 Write	Cycle 1 P0 Read Cycle 2 P1 Read Cycle 3 P0 Write Cycle 4 P0 Read Cycle 5 P0 Write	Cycle 1 P0 Read Cycle 2 P1 Read Cycle 3 P1 Write Cycle 4 P0 Read Cycle 5 P0 Write
(5) SRSW	(6) SRMW	(7) MRSW	(8) MRMW

การจำแนกรูปแบบพื้นที่หน่วยความจำ เป็นขั้นตอนการวิเคราะห์พื้นที่เก็บข้อมูลด้วยการบันทึกจำนวนหน่วยประมวลผลที่เข้าใช้ข้อมูลในพื้นที่นั้นในแต่ละช่วงเวลา โดยในขั้นแรกพี-เทรซจะดึงรายละเอียดการเข้าถึงข้อมูลของโปรแกรมจากซิมิกส์มาเก็บไว้ หลังจากนั้น เมื่อถึงจุดทริกเกอร์ คือเมื่อซิมิกส์ประมวลผลครบรอบเวลา หรือถึงเหตุการณ์ที่ผู้ใช้กำหนดไว้ พี-เทรซจะนำรายละเอียดการเข้าถึงข้อมูลที่เก็บไว้ มาตรวจสอบตามเงื่อนไขเพื่อจำแนกรูปแบบพื้นที่หน่วยความจำ ออกเป็น 8 ชนิด ได้แก่

1. Single Reader (SR) คือ พื้นที่หน่วยความจำที่ถูกหน่วยประมวลผลเพียงหน่วยเดียวอ่านข้อมูล และไม่มีหน่วยประมวลผลอื่นใดเข้าถึงพื้นที่ดังกล่าวในระหว่างนั้น
2. Single Writer (SW) คือ พื้นที่ลักษณะคล้ายกับ SR แต่การเข้าถึงเป็นการเขียน

3. Multiple Readers (MR) คือ พื้นที่หน่วยความจำที่ถูกหลายหน่วยประมวลผลเข้าไปอ่านข้อมูล แต่ไม่มีหน่วยประมวลผลใดเขียนข้อมูลบนพื้นที่ดังกล่าวในระหว่างนั้น

4. Multiple Writers (MW) คือ พื้นที่ลักษณะคล้ายกับ MR แต่การเข้าถึงเป็นการเขียน

5. Single-Reader Single-Writer (SRSW) พื้นที่หน่วยความจำที่ถูกหน่วยประมวลผลตัวหนึ่งเข้าไปเขียนข้อมูล และในขณะเดียวกันมีหน่วยประมวลผลอีกหนึ่งตัว (จะเป็นตัวเดียวกันหรือไม่ก็ได้) เข้าถึงพื้นที่ดังกล่าวเพื่อเขียนข้อมูล

6. Single-Reader Multiple-Writer (SRMW) คือ พื้นที่ลักษณะคล้าย SRSW แต่มีหน่วยประมวลผลอ่าน และหลายหน่วยประมวลผลเขียน

7. Multiple-Reader Single-Writer (MRSW) คือ พื้นที่ลักษณะคล้าย SRSW แต่มีหลายหน่วยประมวลผลอ่าน แต่หนึ่งหน่วยประมวลผลเขียน

8. Multiple-Reader Multiple-Writer (MRMW) คือ พื้นที่หน่วยความจำที่ถูกหลายหน่วยประมวลผลเข้าไปอ่านและเขียนในเวลาเดียวกัน

ภาพที่ 4.4 แสดงตัวอย่างการจำแนกรูปแบบพื้นที่หน่วยความจำเป็นแปดรูปแบบ บนระบบชิปัลติโพรเซสเซอร์ที่มีสองหน่วยประมวลผล คือ P0 และ P1 ในกรอบสี่เหลี่ยมแสดงตัวอย่างรายละเอียดการเข้าถึงข้อมูลในพื้นที่ที่วิเคราะห์ โดยกำหนดทริกเกอร์ให้สรุปข้อมูลเมื่อประมวลผลครบทุก 5 รอบ อักษรปกติเป็นการเข้าถึงข้อมูลของหน่วยประมวลผล P0 และอักษรตัวเข้มเป็นการเข้าถึงข้อมูลของหน่วยประมวลผล P1 ดังรูป(ภาพที่ 4.4 (1 – 4)) หากพื้นที่ที่วิเคราะห์มีการเข้าถึงเพื่ออ่านหรือเขียนเพียงอย่างเดียว จะพิจารณาจำแนกพื้นที่โดยใช้จำนวนหน่วยประมวลผลที่เข้าถึงพื้นที่นั้น เป็น SR, SW, MR หรือ MW แต่หากพื้นที่นั้นมีการเข้าถึงทั้งเพื่ออ่านและเขียนผสมกัน (ภาพที่ 4.4 (5 – 8)) จะจำแนกรูปแบบพื้นที่นั้นโดยนับจำนวนหน่วยประมวลผลที่อ่านและจำนวนหน่วยประมวลผลที่เขียนว่าเป็นหน่วยเดียวหรือหลายหน่วย พื้นที่นั้นจึงถูกจำแนกเป็น SRSW, SRMW, MRSW หรือ MRMW ทั้งนี้ในตัวอย่างของ SRSW หาก P0 เป็นหน่วยเดียวที่อ่าน และ P1 เป็นหน่วยเดียวที่เขียน ในพื้นที่ที่วิเคราะห์ พื้นที่นั้นจะจัดเป็น SRSW เช่นกัน

การจำแนกพื้นที่หน่วยความจำในรูปแบบดังกล่าวจะสะท้อนให้เห็นถึงความถี่การเข้าถึงข้อมูลบนหน่วยความจำในแต่ละช่วงเวลา ซึ่งสามารถใช้อธิบายคุณสมบัติของพื้นที่เก็บข้อมูลของโปรแกรมได้ และแสดงถึงโอกาสที่การใช้ข้อมูลร่วมกันของหลายหน่วยประมวลผลอีกด้วย พื้นที่หน่วยความจำทั้ง 8 รูปแบบสามารถสรุปออกเป็นสองกลุ่มตามการจำแนกของงานวิจัยของมาราธีและมุลเลอร์ (Marathe & Mueller, 2007) ได้แก่ (1) พื้นที่หน่วยความจำที่ไม่ถูกปิดกั้น

(References not protected by locks) คือ พื้นที่หน่วยความจำที่ถูกเขียนจากหน่วยประมวลผลเพียงหน่วยเดียว อาจจะมีหลายหน่วยประมวลผลอ่านก็ได้ คุณลักษณะดังกล่าวนี้จะไม่ทำให้เกิดการแย่งเข้าใช้พื้นที่หน่วยความจำ และ (2) พื้นที่หน่วยความจำที่ถูกปิดกั้น (References protected by locks) คือ พื้นที่หน่วยความจำที่ถูกเขียนจากหลายหน่วยประมวลผล คุณลักษณะดังกล่าวนี้อาจทำให้เกิดการแย่งเข้าใช้พื้นที่หน่วยความจำจากหลายหน่วยประมวลผล ทำให้ระบบปฏิบัติการจำเป็นต้องปิดกั้นพื้นที่และจัดสรรให้เข้าใช้พื้นที่หน่วยความจำครั้งละหนึ่งหน่วยประมวลผลเท่านั้น

แนวทางการจำแนกรูปแบบพื้นที่หน่วยความจำของพี-เทอร์มนี้ ได้ถูกนำไปวิเคราะห์ผลกระทบของคุณสมบัติของพื้นที่เก็บข้อมูลหรือคุณสมบัติการเข้าถึงข้อมูล ที่มีต่อระบบบัสของชิป มัลติโพรเซสเซอร์ พบว่า การประมวลผลโปรแกรมเอฟทีของชุดโปรแกรมเอ็นพีบี บนชิป มัลติโพรเซสเซอร์ที่มี 2 หน่วยประมวลผล หน่วยประมวลผลมีการเข้าถึงพื้นที่หน่วยความจำในระดับเซกเมนต์เป็นแบบ SRSW สูงถึงร้อยละ 69.61 ซึ่งจากรูปแบบดังกล่าวแสดงว่าทั้งสองหน่วยประมวลผลย่อยมีการเข้าถึงข้อมูลในแต่ละเซกเมนต์ในช่วงเวลาที่ไม่ตรงกัน ทำให้ไม่เกิดปัญหาแย่งใช้ข้อมูลชุดเดียวกัน (พีระศักดิ์ วังโสม และ วรวรรณ มะเรงสิทธิ์, 2552ก) และเมื่อศึกษาแนวทางการเพิ่มสมรรถนะโปรแกรมเอฟทีด้วยปรับวิธีการแจกจ่าย พบว่า คำสั่ง **dynamic** ที่ไม่กำหนดขนาดของบล็อกเป็นคำสั่งที่ให้สมรรถนะประมวลผลสูงที่สุด เนื่องจากคำสั่ง **dynamic** ทำให้การเข้าถึงพื้นที่หน่วยความจำแบบ SRMW ลดลง แต่จะมีสัดส่วนการเข้าถึงพื้นที่แบบ MW เพิ่มขึ้น คุณลักษณะดังกล่าวนี้ช่วยให้โปรแกรมมีสมรรถนะสูงขึ้น เนื่องจากการเขียนข้อมูลเพียงอย่างเดียวจากหลายหน่วยประมวลผลทุกหน่วยสามารถเขียนข้อมูลได้ทันที ในขณะที่การเข้าถึงแบบ SRMW นั้น ระหว่างที่หลายหน่วยประมวลผลเขียนข้อมูล หากมีหน่วยใดหน่วยหนึ่งต้องการอ่านข้อมูล แคมป์จะต้องมีวิธีการจัดการให้ข้อมูลที่มีอยู่ในแต่ละหน่วยประมวลผลมีความสอดคล้องกัน (พีระศักดิ์ วังโสม และ วรวรรณ มะเรงสิทธิ์, 2552ข)

ภาพที่ 4.5 โครงสร้างการบันทึกข้อมูลการเข้าถึงหน่วยความจำลงบนเทอร์ซไฟล์ของพี-เทอร์ม

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

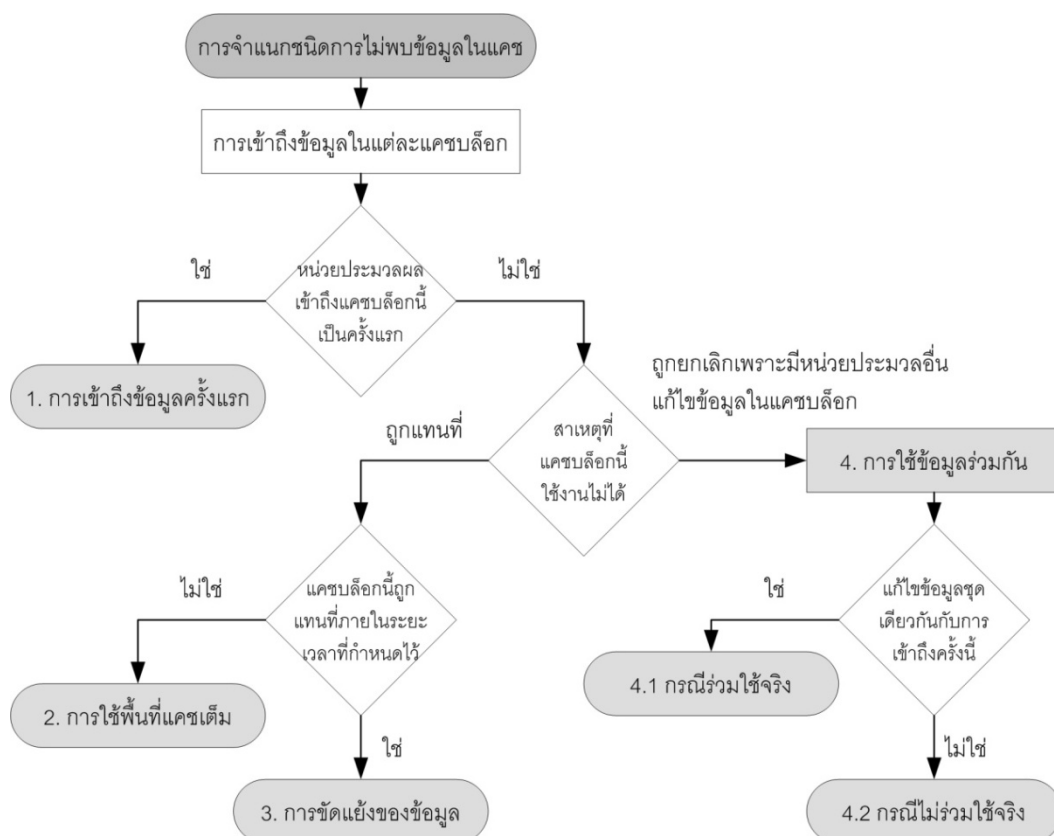
ภายหลังจากจำแนกรูปแบบพื้นที่หน่วยความจำแล้ว จะมีการบันทึกข้อมูลลงบนเทอร์ซไฟล์ ซึ่งเป็นขั้นตอนการสรุปข้อมูลรูปแบบพื้นที่ที่จำแนกแล้ว เป็นระเบียบเพื่อเขียนลงในเทอร์ซไฟล์ โครงสร้างการบันทึกข้อมูลดังแสดงในภาพที่ 4.5 ซึ่งแต่ละระเบียบจะประกอบด้วย ลำดับการเข้าถึงข้อมูล หน่วยประมวลผล ชนิดการเข้าถึง และตำแหน่งพื้นที่เก็บข้อมูลบนหน่วยความจำ เมื่อ

โปรแกรมประมวลผลเสร็จเรียบร้อยแล้ว ไฟล์ดังกล่าวจะถูกส่งต่อไปยังพี-กราฟเพื่อใช้เป็นข้อมูลให้โปรแกรมกนุพลอดนำไปสร้างกราฟต่อไป

4.2.2 ส่วนการจำแนกชนิดการไม่พบข้อมูลในแคช

ดังที่กล่าวไว้ในหัวข้อที่ 4.1.1.2 ซิมิกส์มีโมดูลจี-แคชทำหน้าที่เป็นแคชเสมือน ทำหน้าที่รับรายการ (Transaction) การเข้าถึงข้อมูลของหน่วยประมวลผล เพื่อนำมาจำแนกว่าข้อมูลดังกล่าวมีอยู่ในแคชในขณะนั้นหรือไม่ หากไม่พบข้อมูลจี-แคชส่งรายการดังกล่าวไปยังจี-แคชที่อยู่ระดับถัดไปหรือส่งไปยังหน่วยความจำหลัก เพื่อดึงข้อมูลส่งกลับไปที่หน่วยประมวลผล เมื่อการประมวลผลโปรแกรมเสร็จสิ้น ผู้ใช้สามารถเรียกดูข้อมูลของจำนวนรายการเข้าถึงข้อมูลทั้งหมด และอัตราการพบข้อมูลจากจี-แคชได้ จี-แคชมีค่าพารามิเตอร์ให้ผู้ใช้ปรับแต่งได้หลายค่า ได้แก่ ชนิดและขนาดของแคช ขนาดแคชบล็อก จำนวนเซต และนโยบายการเขียนข้อมูล สำหรับกลไกการรักษาความสอดคล้องของข้อมูลบนแคช ซิมิกส์เวอร์ชันปัจจุบันรองรับเฉพาะโปรโตคอลเมชีเท่านั้น

ภาพที่ 4.6 อัลกอริทึมการจำแนกชนิดการไม่พบข้อมูลในแคชของพี-แคช



แม้ว่าจี-แคชจะสามารถนำเสนอข้อมูลอัตราการพบข้อมูลและค่าพารามิเตอร์ที่ครบถ้วน แต่ทว่าข้อมูลที่จี-แคชจัดเก็บยังไม่เพียงพอที่จะนำมาจำแนกชนิดการไม่พบข้อมูลในแคชได้ งานวิจัยนี้จึงได้พัฒนาโมดูลพี-แคชให้ทำงานร่วมกับจี-แคช เพื่อรับรายการเข้าถึงที่ไม่พบข้อมูลในแคชมาจำแนก ซึ่งแบ่งออกเป็นสี่ชนิด ประกอบด้วย (1) การเข้าถึงข้อมูลครั้งแรก (2) การใช้พื้นที่แคชเต็ม (3) การขัดแย้งของข้อมูล และ (4) การใช้ข้อมูลร่วมกัน (กรณีร่วมใช้จริงและไม่จริง)

พี-แคชใช้อัลกอริทึมในภาพที่ 4.6 ในการจำแนกชนิดการไม่พบข้อมูล โดยพิจารณาว่ารายการดังกล่าวเป็นการเข้าถึงข้อมูลในแคชบล็อกครั้งแรกหรือไม่ หากใช่ก็จัดเป็นการไม่พบข้อมูลเนื่องจากเข้าถึงข้อมูลครั้งแรก ในกรณีที่ไม่ใช่จะพิจารณาต่อไปว่าสาเหตุใดที่ทำให้แคชบล็อกดังกล่าวไม่สามารถใช้งานได้ หากแคชบล็อกดังกล่าวถูกแทนที่ไปและถูกเข้าถึงอีกครั้งภายในระยะเวลาที่กำหนดไว้ (Reuse distance) แสดงว่าหน่วยประมวลผลมีการเข้าถึงข้อมูลที่ถูกเก็บไว้บนแคชบล็อกเดียวกันบ่อยครั้ง ซึ่งเป็นการไม่พบข้อมูลเนื่องจากเกิดการขัดแย้งของข้อมูล (ผู้ใช้สามารถกำหนดช่วงระยะเวลาการถูกแทนที่จนกระทั่งข้อมูลถูกเข้าถึงอีกครั้งได้) ส่วนแคชบล็อกที่ถูกแทนที่ไปเป็นเวลานานกว่าระยะเวลาที่กำหนดไว้ ก็จัดเป็นการไม่พบข้อมูลเนื่องจากการใช้พื้นที่แคชเต็ม ส่วนในกรณีที่แคชบล็อกถูกยกเลิกเพราะมีหน่วยประมวลผลอื่นแก้ไขข้อมูลในแคชบล็อกดังกล่าว ทำให้แคชต้องใช้กลไกเพื่อรักษาความสอดคล้องของข้อมูลโดยการยกเลิกแคชบล็อกดังกล่าวไม่ให้อ่านได้ การไม่พบข้อมูลที่เกิดขึ้นจัดเป็นการไม่พบข้อมูลเนื่องจากใช้ข้อมูลร่วมกัน ซึ่งถ้าหากการแก้ไขข้อมูลเป็นข้อมูลชุดเดียวกับการเข้าถึงข้อมูลจะจัดเป็นกรณีร่วมใช้จริง แต่ถ้าหากเป็นข้อมูลคนละชุดจะจัดเป็นกรณีไม่ร่วมใช้จริง

ภาพที่ 4.7 โครงสร้างข้อมูลที่ใช้เก็บประวัติและสถานะของพื้นที่ข้อมูลที่ถูกรับเข้าถึง

[Set]	[Set status]	[Address]	[Access count]	[Current time]	[Last time]
(1)	(2)	(3)	(4)	(5)	(6)
[Access byte] [Coherence byte] [Coherence status]					
(7)		(8)	(9)		

เพื่อให้โมดูลพี-แคช สามารถจำแนกชนิดการไม่พบข้อมูลตามอัลกอริทึมดังกล่าวได้ พี-แคชถูกออกแบบให้มีโครงสร้างข้อมูลเพื่อใช้เก็บประวัติการเข้าถึงและสถานะของพื้นที่ข้อมูล (Memory address) ในแต่ละพื้นที่ไว้ ดังในภาพที่ 4.7 ซึ่งประกอบด้วย (1) เซต หมายถึง เซตของแคชที่พื้นที่ข้อมูลดังกล่าวจะถูกดึงเข้าไปเมื่อถูกเข้าถึง (2) สถานะของเซต ใช้ตรวจสอบว่าเซตดังกล่าวนี้มีพื้นที่ว่างเพื่อเก็บข้อมูลอีกหรือไม่ (3) พื้นที่ข้อมูล ใช้เก็บตำแหน่งพื้นที่ข้อมูล (4)

จำนวนการเข้าถึง ใช้เก็บจำนวนการเข้าถึงพื้นที่ข้อมูล (5) เวลา (Timestamp) ของการเข้าถึง ปัจจุบัน (6) เวลาของการเข้าถึงครั้งสุดท้าย (7) ตำแหน่งไบนารีของพื้นที่ข้อมูลที่ถูกเข้าถึง (8) ตำแหน่งไบนารีที่ถูกยกเลิกเนื่องจากการแก้ไขข้อมูลของหน่วยประมวลผลอื่น และ (9) สถานะของพื้นที่ข้อมูลเพื่อใช้ตรวจสอบกลไกการรักษาความสอดคล้องของข้อมูล

จากโครงสร้างข้อมูลประวัติและสถานะดังกล่าว จำนวนการเข้าถึงจะใช้ในการตรวจสอบเป็นการเข้าถึงข้อมูลครั้งแรกหรือไม่ สถานะของเซตใช้ตรวจสอบเป็นการไม่พบข้อมูลเนื่องจากพื้นที่แคชเต็มหรือไม่ และเวลาการเข้าถึงปัจจุบันและครั้งสุดท้ายใช้ตรวจสอบการขัดแย้งของข้อมูล ในขณะที่ตำแหน่งไบนารีที่ถูกเข้าถึงและถูกยกเลิก รวมถึงสถานะของกลไกการรักษาความสอดคล้องของข้อมูล จะถูกใช้ตรวจสอบการไม่พบข้อมูลเนื่องจากการใช้ข้อมูลร่วมกัน หากตำแหน่งไบนารีที่ถูกเข้าถึงและถูกยกเลิกเป็นตำแหน่งเดียวกันจะจัดเป็นกรณีร่วมใช้จริง หากไม่ใช่ก็จัดเป็นกรณีร่วมใช้ไม่จริง ซึ่งข้อมูลชนิดของการไม่พบข้อมูลนี้เป็นความสามารถของพี-สเปาที่พัฒนาเพิ่มเติมบนระบบจำลองซิมิกซ์

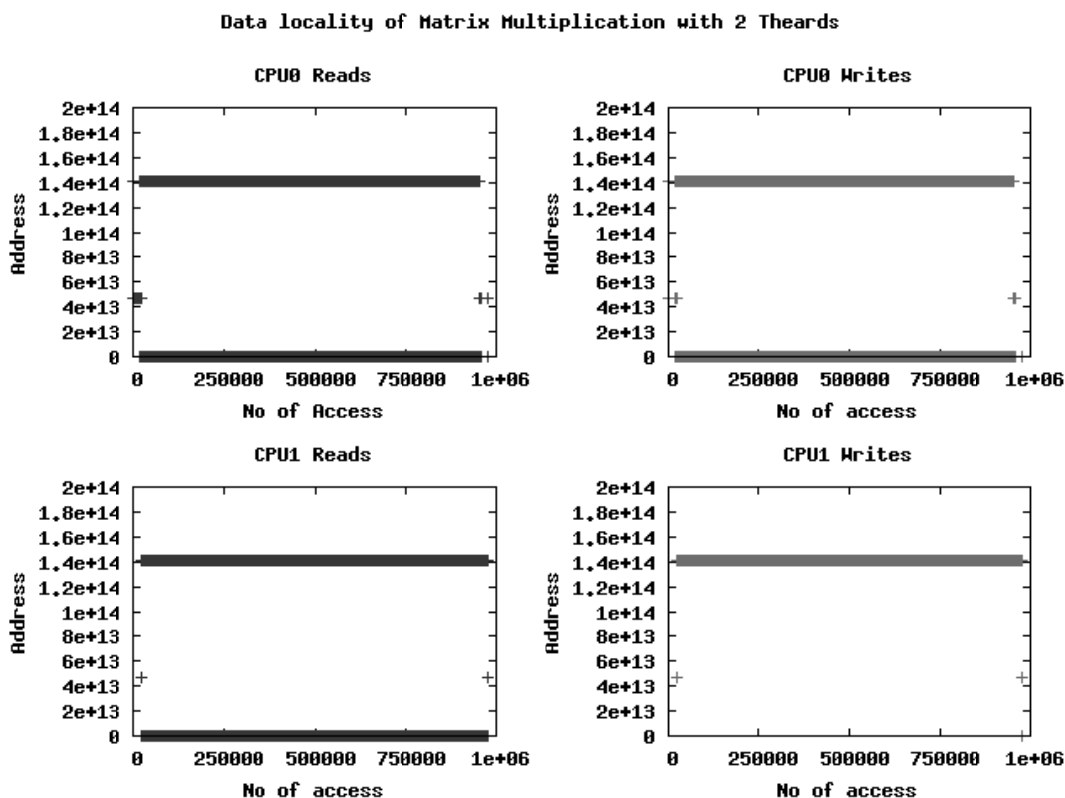
4.2.3 ส่วนการนำเสนอข้อมูลในรูปแบบกราฟ

ส่วนการทำงานสุดท้ายคือส่วนการนำเสนอข้อมูลในรูปแบบของกราฟ งานวิจัยนี้ได้พัฒนาโมดูลพี-กราฟเพื่อเป็นทางเลือกให้ผู้ใช้สามารถเรียกดูข้อมูลในลักษณะของกราฟได้ เนื่องจากปัจจุบันซิมิกซ์นำเสนอข้อมูลในรูปแบบของข้อความเพียงอย่างเดียว พี-กราฟเป็นส่วนประสานงานกับโปรแกรมกนูพล็อต (Gnuplot) (Gnuplot, 2007) ซึ่งเป็นโปรแกรมที่ใช้สร้างกราฟภายหลังที่โปรแกรมประมวลผลและพี-สเปาบันทึกการทำงานลงเทรซไฟล์เป็นที่เรียบร้อย พี-กราฟจะสร้างไฟล์บทคำสั่งให้กับโปรแกรมกนูพล็อต เพื่อนำเทรซไฟล์ที่ได้ไปสร้างกราฟต่อไป

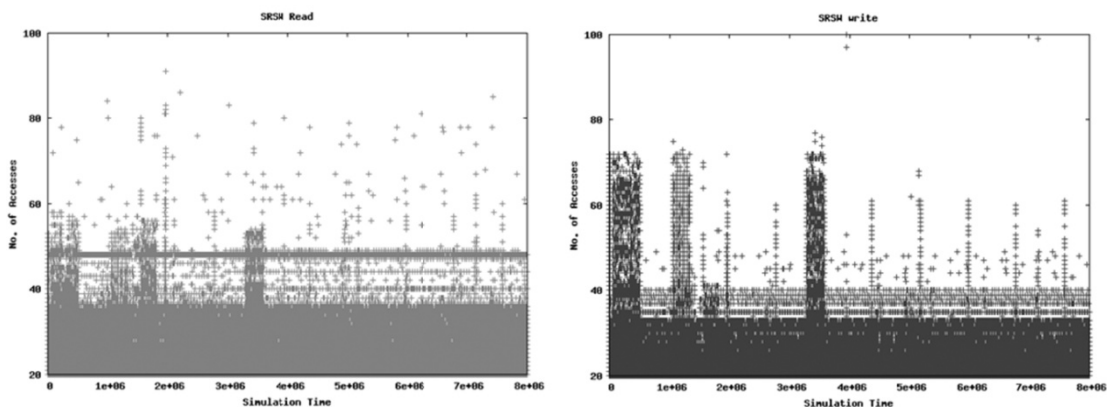
ข้อมูลในเทรซไฟล์เป็นรายละเอียดการเข้าถึงข้อมูลของหน่วยประมวลผลก่อนเข้าถึงแคช ข้อมูลดังกล่าวสามารถนำมาสร้างกราฟได้สองลักษณะ (ภาพที่ 4.8) ประกอบด้วย ตัวอย่างแรก (ภาพที่ 4.8 (a)) เป็นกราฟผลการวิเคราะห์คุณสมบัติพื้นที่เก็บข้อมูลของโปรแกรมการคูณเมทริกซ์แบบขนาน ซึ่งเป็นส่วนหนึ่งของการทดลองในบทที่ 5.3 ในแต่ละกราฟย่อยจะแยกตามชนิดการเข้าถึง (อ่านและเขียน) ของแต่ละหน่วยประมวลผล โดยมีแกนแนวนอนเป็นลำดับในการเข้าถึงข้อมูล และแกนแนวตั้งแสดงตำแหน่งพื้นที่ข้อมูลที่ถูกเข้าถึง จากลักษณะของกราฟจะช่วยอธิบายถึงความหนาแน่นของการเข้าถึงของพื้นที่ข้อมูลในแต่ละตำแหน่ง จากภาพที่ 4.8 (a) พื้นที่ข้อมูลถูกเข้าถึงบ่อยครั้งอย่างต่อเนื่องหรือมีคุณสมบัติพื้นที่เก็บข้อมูลที่ถูกใช้บ่อยครั้ง

ภาพที่ 4.8 ตัวอย่างการนำเสนอข้อมูลในรูปแบบของกราฟของพี-สเป

(a) คุณสมบัติการใช้พื้นที่เก็บข้อมูลของโปรแกรมการคูณเมทริกซ์ บนชิปมัลติโพรเซสเซอร์ขนาด 2 หน่วย



(b) จำนวนการเข้าถึงข้อมูลบนเซกเมนต์รูปแบบ SRSW ของโปรแกรมเอฟที



(b.1) จำนวนการอ่าน

(b.2) จำนวนการเขียน

ตัวอย่างที่สอง (ภาพที่ 4.8 (b)) เป็นกราฟผลการวิเคราะห์รูปแบบพื้นที่หน่วยความจำของโปรแกรมเอฟที ซึ่งเป็นส่วนหนึ่งของงานวิจัยศึกษาผลกระทบของคุณสมบัติของพื้นที่เก็บข้อมูลที่มิติต่อระบบบัสของชิปมัลติโพรเซสเซอร์ (พีระศักดิ์ วังโสม และ วรวรรณ มะเวงสิทธิ์, 2552ก)

กราฟดังกล่าวถูกสร้างขึ้นหลังจากที่พี-สปามีจำแนกรูปแบบพื้นที่หน่วยความจำเสร็จสิ้น หากผู้ใช้ต้องการวิเคราะห์ข้อมูลพื้นที่ในแต่ละรูปแบบก็สามารถสั่งให้พี-กราฟสร้างของแต่ละพื้นที่ได้ ในตัวอย่างเป็นจำนวนการเข้าถึงของโปรแกรมเอฟทีบนพื้นที่หน่วยความจำแบบ SRSW ซึ่งเป็นพื้นที่โดยส่วนใหญ่ของโปรแกรม โดยแกนแนวนอนคือลำดับการเข้าถึงในแต่ละช่วง ซึ่งผู้ใช้สามารถกำหนดช่วงที่ต้องการศึกษาด้วยเวลาหรือเหตุการณ์ได้ เช่น เก็บข้อมูลเมื่อประมวลผลครบ 100 รอบสัญญาณนาฬิกา หรือเก็บข้อมูลเมื่อมีการเข้าถึงข้อมูลครบ 100 ครั้ง ซึ่งแกนแนวดิ่งจะแสดงถึงปริมาณการเข้าถึงข้อมูลในแต่ละช่วง จากลักษณะของกราฟจะแสดงถึงความหนาแน่นของการเข้าถึงพื้นที่ข้อมูล หากพื้นที่หน่วยความจำที่ถูกเขียนจากหลายหน่วยประมวลผล มีความหนาแน่นของการเขียนสูง ก็จะทำให้โปรแกรมมีโอกาสสูญเสียสมรรถนะการประมวลผลได้

4.2.4 สรุปส่วนการทำงานเพิ่มเติมของพี-สปา

เครื่องมือวิเคราะห์พี-สปาถูกพัฒนาขึ้นมาโดยมีวัตถุประสงค์เพื่อให้นำเสนอข้อมูลการใช้ประโยชน์พื้นที่แคช ที่มีผลจากการประมวลผลของในรูปของโปรแกรมแบบโอเพนเอ็มพี พี-สปาจึงถูกพัฒนาเพิ่มเติมบนระบบจำลองซิมิกส์ เพื่อให้สามารถวิเคราะห์และนำเสนอข้อมูลที่เพิ่มเติมจากซิมิกส์ได้สามลักษณะ คือ

1. การวิเคราะห์คุณสมบัติพื้นที่เก็บข้อมูล พี-สปามีพี-เทรซเป็นส่วนการทำงานที่ใช้วิเคราะห์คุณสมบัติพื้นที่เก็บข้อมูล ซึ่งได้อธิบายไว้ในหัวข้อที่ 4.2.1 พี-เทรซสามารถปรับการวิเคราะห์พื้นที่หน่วยความจำได้ตั้งแต่ระดับไบต์ แคชบล็อก เพจ จนถึงระดับเซกเมนต์ ตามความเหมาะสมของสถาปัตยกรรมระบบคอมพิวเตอร์เป้าหมาย

2. การจำแนกชนิดการไม่พบข้อมูลในแคช ซึ่งพี-สปามีพี-แคชทำหน้าที่จำแนกดังกล่าวรายละเอียดไว้ในหัวข้อที่ 4.2.2 การจำแนกชนิดการไม่พบข้อมูลของพี-แคชสามารถจำแนกชนิดได้สี่ชนิด ประกอบด้วย (1) การเข้าถึงข้อมูลครั้งแรก (2) การใช้พื้นที่แคชเต็ม (3) การขัดแย้งของข้อมูล และ (4) การใช้ข้อมูลร่วมกัน ซึ่งชนิดสุดท้ายนี้ยังสามารถจำแนกออกเป็นกรณีที่ใช้ข้อมูลร่วมกันจริงและกรณีที่ใช้ข้อมูลร่วมกันไม่จริงอีกด้วย

3. การนำเสนอข้อมูลในรูปแบบของกราฟ เพื่อเป็นทางเลือกในการนำเสนอข้อมูลให้แก่ผู้ใช้ พี-สปามีพี-กราฟทำหน้าที่สร้างกราฟโดยนำข้อมูลจากเทรซไฟล์ที่ได้จากการประมวลผลมาสรุปข้อมูลในรูปแบบของกราฟ เพื่อให้ผู้ใช้ทำความเข้าใจได้เร็วขึ้น

ภายหลังที่พัฒนาพี-สไปให้สามารถนำเสนอข้อมูลได้ครบถ้วนตามวัตถุประสงค์ของงานแล้ว ขั้นตอนถัดไปเป็นการทดสอบความถูกต้องของการทำงานของเครื่องมือวิเคราะห์พี-สไป ซึ่งจะได้กล่าวถึงรายละเอียดการทดสอบในหัวข้อถัดไป

4.3 การทดสอบความถูกต้องของเครื่องมือวิเคราะห์พี-สไป

ก่อนที่จะนำพี-สไปไปใช้เป็นเครื่องมือเพื่อศึกษาสมรรถนะของโปรแกรมบนชิป มัลติโพรเซสเซอร์ จำเป็นต้องมีการทดลองเพื่อยืนยันว่าผลลัพธ์ที่เก็บจากพี-สไปนั้นถูกต้องและน่าเชื่อถือ ภายหลังจากการพัฒนาเครื่องมือแต่ละส่วนของพี-สไปเสร็จเรียบร้อยแล้วนั้น จึงมีการทดลองเพื่อนำข้อมูลที่ได้จากพี-สไปมาเปรียบเทียบกับข้อมูลที่ได้จากการประมวลผลบนระบบคอมพิวเตอร์จริง ข้อมูลที่นำมาเปรียบเทียบมีสองส่วน คือ (1) เวลาประมวลผล และ (2) อัตราการไม่พบข้อมูลในแคชระดับที่ 2 งานวิจัยนี้ได้ทดสอบความถูกต้องบนเครื่องคอมพิวเตอร์ที่ใช้ชิป มัลติโพรเซสเซอร์ของบริษัทอินเทล² ที่มี 2 หน่วยประมวลผล ซึ่งเป็นหน่วยประมวลผลที่ใช้กันอย่างแพร่หลายในปัจจุบัน โดยใช้ชุดโปรแกรมเอ็นพีบี 2.3 (Jin *et al.*, 1999) ขององค์นาซ่าที่พัฒนาด้วยโอเพนเอ็มพีบนภาษาซีเป็นโปรแกรมภาระงานเพื่อประมวลผล

4.3.1 คุณลักษณะของโปรแกรมเอ็นพีบี 2.3

ชุดโปรแกรมเอ็นพีบี 2.3 ประกอบด้วย 8 โปรแกรมย่อย แบ่งตามลักษณะการประมวลผลเป็นสองกลุ่ม คือ กลุ่มโปรแกรมจำลองการไหลของสารเคมี (Simulated chemical fluid dynamic) และกลุ่มภาระงานหลักของการคำนวณทางวิทยาศาสตร์ ในกลุ่มโปรแกรมแรกมีสามโปรแกรม ได้แก่ โปรแกรมบีที (BT) แอลยู (LU) และเอสพี (SP) ส่วนกลุ่มโปรแกรมที่สองมีห้าโปรแกรม ได้แก่ ซีจี (CG) อีพี (EP) เอฟที (FT) ไอเอส (IS) และเอ็มจี (MG) แต่ละโปรแกรมสามารถกำหนดขนาดของข้อมูลในการประมวลผลได้ห้าคลาส (W, S, A, B และ C เรียงลำดับตามขนาดของข้อมูลที่ใช้เพื่อทดสอบการประมวลผลของเครื่องระดับเล็กไปจนถึงขนาดข้อมูลขนาดใหญ่สำหรับทดสอบเครื่องสมรรถนะสูง)

² อินเทล คอร์ 2 ดูโอ รุ่นที่ 6800

โปรแกรมย่อยทั้งแปดโปรแกรมมีคุณลักษณะโดยสรุป ดังนี้

1. โปรแกรมบีที (BT) เป็นโปรแกรมจำลองการไหลของสารเคมี ซึ่งเป็นการจำลองด้วยวิธีการแก้สมการแบบนาเวียร์-สโตกส์ (Navier-stokes equation) โดยผลลัพธ์จะอยู่ในรูปของเมทริกซ์สามแนวเฉียง (Tridiagonal) ขนาด 5x5 บล็อก
2. โปรแกรมแอลยู (LU) เป็นโปรแกรมจำลองการไหลของสารเคมี ด้วยวิธีผ่อนปรนส่วนเกินสลับเนื่องแบบเป็นระบบ (SSOR - Systematic symmetric successive over-relaxation) ซึ่งเป็นการแบ่งเมทริกซ์ออกเป็นสามบนและล่างตามแนวเฉียง (Lower and upper triangular)
3. โปรแกรมเอสพี (SP) เป็นโปรแกรมจำลองการไหลของสารเคมี ด้วยวิธีการแก้สมการแบบบีม-วอร์มมิง (Beam-warming) บนชุดข้อมูลสามมิติ
4. โปรแกรมซีจี (CG) เป็นโปรแกรมการคำนวณคอนจูเกต เกรเดียนต์ (Conjugate gradient) ซึ่งเป็นการคำนวณหาค่าลักษณะเฉพาะ (Eigenvalue) ของเมทริกซ์ขนาดใหญ่ที่มีค่าว่าง (Large sparse matrix)
5. โปรแกรมอีพี (EP) เป็นโปรแกรมที่สามารถประมวลผลแบบขนานได้ง่าย (Embarrassingly parallel program) ซึ่งเป็นการสร้างค่าตัวเลขสุ่มแบบเกาส์เซียน (Gaussian random)
6. โปรแกรมเอฟที (FT) เป็นโปรแกรมคำนวณการแปลงฟูเรียร์แบบเร็วบนชุดข้อมูลสามมิติ (3-D Fast Fourier transform) ด้วยวิธีการแบบสเปกตรัล (Spectral method)
7. โปรแกรมไอเอส (IS) เป็นโปรแกรมเรียงลำดับชุดตัวเลขจำนวนเต็มขนาดใหญ่ (Large integer sorting)
8. โปรแกรมเอ็มจี (MG) เป็นโปรแกรมคำนวณวิธีกริดหลายระดับ (Multigrid method) ด้วยวัฏจักรวี (V-cycle) ซึ่งเป็นการคำนวณเพื่อแก้สมการพอยส์ซันของชุดข้อมูลสเกลาร์สามมิติ (3-D scalar poisson equation)

4.3.2 วิธีทำการทดลอง

การทดลองเพื่อทดสอบความถูกต้องนี้ได้ประมวลผลชุดโปรแกรมเอ็นพีบี 2.3 คลาสเอสด้วยจำนวน 2 เทรด บนชิปัลติโพรเซสเซอร์ดังแสดงในตารางที่ 4.2 การเก็บข้อมูลบนระบบคอมพิวเตอร์จึงแบ่งออกเป็นสองส่วน คือ การเก็บเวลาประมวลผล ได้เก็บเวลาประมวลผลจากคำสั่ง **time** ของระบบปฏิบัติการลินุกซ์ มีหน่วยเป็นวินาที และการเก็บข้อมูลอัตราการใช้

ข้อมูลในแคชระดับที่ 2 ได้เก็บข้อมูลจากเครื่องมือวิเคราะห์สมรรถนะวีทูน (Vtune performance analyzer) ของบริษัทอินเทล บนระบบปฏิบัติการวินโดวส์ ส่วนระบบจำลองนั้น ได้ประมวลผลโปรแกรมเอ็นพีบีบนระบบปฏิบัติการลินุกซ์ และใช้พี-สเปเก็บข้อมูลทั้งสองส่วน โดยเวลาประมวลผลใช้หน่วยจำนวนรอบสัญญาณนาฬิกาที่ใช้ประมวลผล และหารด้วยความเร็วรอบสัญญาณนาฬิกาของระบบคอมพิวเตอร์จริง เพื่อแปลงหน่วยเป็นวินาที ขั้นตอนสุดท้ายคือการนำข้อมูลที่ได้จากการทดลองทั้งสองส่วนไปวิเคราะห์ความถดถอย (Regression analysis) ซึ่งการทดลองนี้มีสมมติฐานคือ เวลาประมวลผลและอัตราการไม่พบข้อมูลบนแคชระดับที่ 2 ที่ได้จากพี-สเป สามารถนำไปใช้ทำนายหรือเป็นตัวแทนของข้อมูลที่ได้จากการประมวลผลบนระบบคอมพิวเตอร์จริงได้

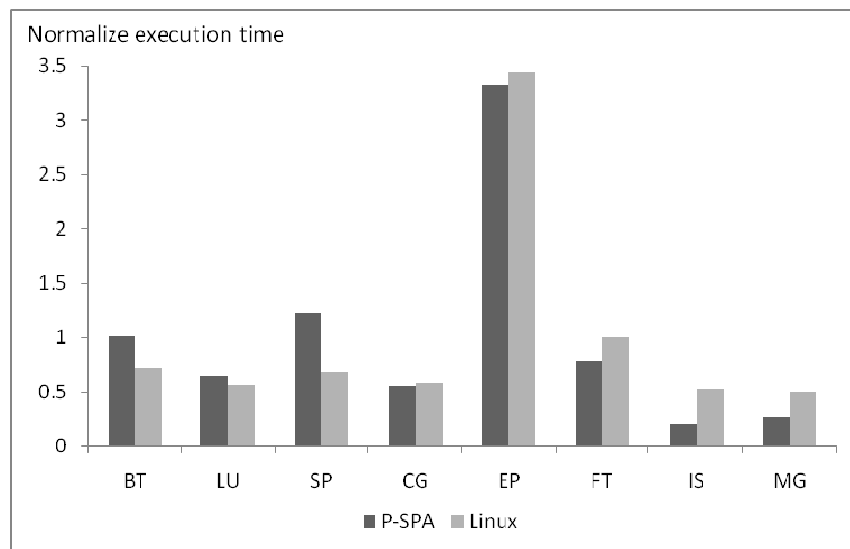
ตารางที่ 4.2 สถาปัตยกรรมของระบบคอมพิวเตอร์ที่ใช้ทดสอบความถูกต้อง

สถาปัตยกรรมของระบบคอมพิวเตอร์	ค่าทางสถาปัตยกรรม
สถาปัตยกรรมของหน่วยประมวลผล	
- จำนวนหน่วยประมวลผล	2 หน่วย
- ความเร็วสัญญาณนาฬิกา	2.0 GHz
- ชุดคำสั่ง	x86_64
สถาปัตยกรรมของแคชระดับที่ 1	
- ขนาดของแคชระดับที่ 1 (แยกแชนจ์คำสั่งและข้อมูล)	32 KB / private
- ขนาดของแคชบล็อก	64 bytes
- จำนวนเซต	8 way
- นโยบายการเขียน	write-back, allocate
- เวลาการเข้าถึงข้อมูล	3 cycles
สถาปัตยกรรมของแคชระดับที่ 2	
- ขนาดของแคชระดับที่ 2	4 MB / Shared
- ขนาดของแคชบล็อก	64 bytes
- จำนวนเซต	16 way
- นโยบายการเขียน	write-back, allocate
- เวลาการเข้าถึงข้อมูล	10 cycles
- เวลาการเข้าถึงข้อมูลในกรณีไม่พบข้อมูล	165 cycles
ความจุของหน่วยความจำหลัก	4 GB

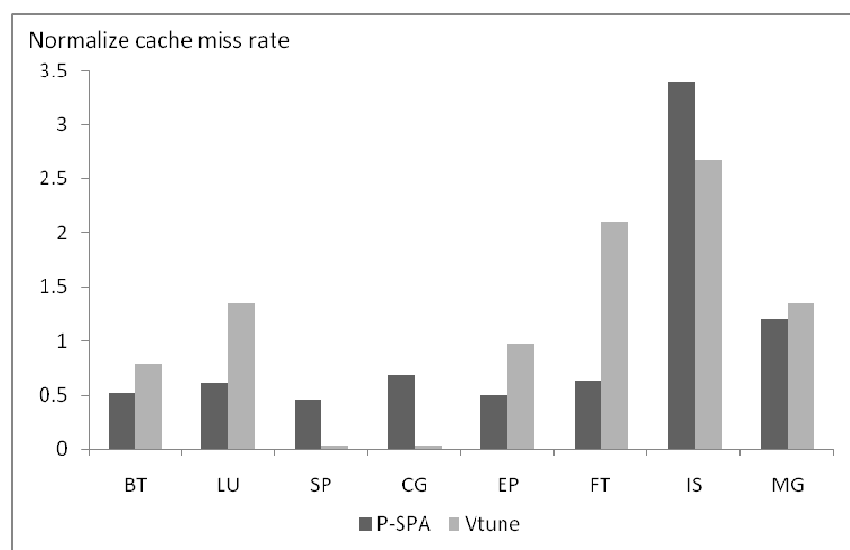
4.3.3 ผลการทดลอง

ผลการทดลอง (ภาพที่ 4.9) โดยในภาพที่ 4.9 (a) เป็นกราฟของเวลาประมวลผล และภาพที่ 4.9 (b) เป็นกราฟของอัตราการไม่พบข้อมูลในแคชระดับที่ 2 ของพี-สเปและระบบคอมพิวเตอร์จริง ซึ่งข้อมูลที่แสดงในกราฟทั้งสองกราฟเป็นข้อมูลที่ได้ปรับให้อยู่เป็นบรรทัดฐานเดียวกัน (Normalize)

ภาพที่ 4.9 ผลการทดลองเพื่อทดสอบความถูกต้องของเครื่องมือวิเคราะห์พี-สเป
(a) เวลาประมวลผลของพี-สเปและระบบคอมพิวเตอร์จริง บนชุดโปรแกรมเอ็นพีบี 2.3



(b) อัตราการไม่พบข้อมูลในแคชระดับที่ 2 ของพี-สเปและระบบคอมพิวเตอร์จริง บนชุดโปรแกรมเอ็นพีบี 2.3



จากผลการวิเคราะห์ความถดถอยของเวลาประมวลผล พบว่า เวลาประมวลผลของพี-สแปมีความสัมพันธ์กับเวลาประมวลผลของระบบคอมพิวเตอร์จริง อย่างมีนัยสำคัญ ($R = 0.957, p \leq 0.05$) และผลการวิเคราะห์ความถดถอยของอัตราการไม่พบข้อมูลในแคชระดับที่ 2 พบว่า อัตราการไม่พบข้อมูลของพี-สแปมีความสัมพันธ์กับอัตราการไม่พบข้อมูลของระบบคอมพิวเตอร์จริง อย่างมีนัยสำคัญ ($R = 0.714, p \leq 0.05$)

4.3.4 สรุปผลการทดสอบความถูกต้อง

จากผลการวิเคราะห์ทางสถิติจะเห็นว่า ข้อมูลเกี่ยวกับเวลาในการประมวลผลที่เก็บค่าจากเครื่องมือพี-สแปมีลักษณะที่สอดคล้องกับระบบจริงมากกว่าข้อมูลเกี่ยวกับอัตราการไม่พบข้อมูล ซึ่งปัจจัยหนึ่งที่ทำให้ค่าผลลัพธ์ของอัตราการไม่พบข้อมูลมีความแตกต่างสูงนั้นเนื่องจากระบบเป้าหมายที่กำหนดในพี-สเปานั้นใช้ระบบปฏิบัติการลินุกซ์ แต่การวิเคราะห์ค่าจากระบบจริงนั้นเก็บจากโปรแกรมวิทูนซึ่งทำงานบนระบบปฏิบัติการวินโดวส์ ทั้งนี้ในระหว่างทำการทดลองได้สังเกตว่า ลินุกซ์มีแนวโน้มที่จะกำหนดเทรตให้กับหน่วยประมวลผลเพียงคอร์เดียว หรือให้กับคอร์เดียวมากกว่า ทำให้เมื่อพี-สแปเก็บข้อมูลของเทรตที่สำเนาไว้ที่แคชจากระบบจำลองซิมิกซ์ที่กำหนดค่าเป็นลินุกซ์จึงทำให้ค่าที่เก็บได้แตกต่างจากค่าที่วัดจากระบบวินโดวส์เป็นอย่างมาก

ในแอปดโปรแกรมที่นำมาวิเคราะห์พบว่าอัตราการไม่พบข้อมูลของโปรแกรมเอฟฟี่ ที่เก็บได้จากพี-สเปานั้นมีความแตกต่างจากค่าที่เก็บจากระบบจริงมากที่สุด ซึ่งสันนิษฐานได้ว่าโปรแกรมเอฟฟี่มีจำนวนครั้งในการเรียกใช้ข้อมูลสูง และมีลักษณะการร่วมใช้ข้อมูลในลักษณะที่หน่วยหนึ่งอ่านและหน่วยหนึ่งเขียนสูงมากเมื่อเทียบกับโปรแกรมอื่น จึงทำให้หากกำหนดเทรตให้หน่วยประมวลผลที่อ่านและเขียนข้อมูลอยู่ที่เดียวกันดังที่ระบบปฏิบัติการลินุกซ์มักจะกำหนดให้หลายเทรตไปร่วมใช้หน่วยประมวลผลเดียวกัน มีผลให้สำเนาข้อมูลในแคชถูกใช้ซ้ำ อัตราการไม่พบข้อมูลจึงต่ำกว่าค่าที่จับได้จากระบบจริงซึ่งใช้ระบบปฏิบัติการวินโดวส์เป็นอย่างมาก อย่างไรก็ตาม อัตราการไม่พบข้อมูลในโปรแกรม เอฟฟี่นั้นมีสัดส่วนน้อยมาก (เพียงร้อยละ 0.03) จึงทำให้เวลาสูญเสียไปจากการไม่พบข้อมูลนั้นมีผลต่อเวลาในการประมวลผลน้อยมาก ดังจะเห็นในภาพที่ 4.9 (a) ว่าเวลาในการประมวลผลของโปรแกรมเอฟฟี่ที่จับได้จากพี-สแปและเครื่องจริงไม่ได้มีความแตกต่างสูงนัก

จากเป้าหมายการทดลองเพื่อทดสอบความถูกต้องของเครื่องมือวิเคราะห์พี-สแป เพื่อยืนยันว่า ข้อมูลที่ได้จากพี-สแปสามารถนำไปใช้ทำนายหรือเป็นตัวแทนของข้อมูลที่ประมวลผลบน

ระบบคอมพิวเตอร์จริงได้ จากการวิเคราะห์ความถดถอยของเวลาประมวลผลและอัตราการไม่พบข้อมูลในแคชระดับที่ 2 บนพี-สแปเทียบกับระบบคอมพิวเตอร์จริง พบว่า ข้อมูลทั้งสองส่วนที่ได้จากพี-สแปมีความสัมพันธ์กับข้อมูลจากระบบคอมพิวเตอร์อย่างมีนัยสำคัญ จึงสามารถสรุปผลการทดสอบถูกต้องได้ว่า พี-สแปสามารถนำเสนอข้อมูลที่สามารถใช้ทำนายหรือเป็นตัวแทนของการประมวลผลบนระบบคอมพิวเตอร์จริงเป็นที่ยอมรับได้ งานวิจัยนี้จะได้นำเครื่องมือวิเคราะห์พี-สแปไปใช้วิเคราะห์สมรรถนะของโปรแกรมแบบโอเพนเอ็มพี เพื่อแสดงถึงศักยภาพของพี-สแปซึ่งกล่าวถึงในบทที่ 5 และ 6

4.4 สรุปวิธีการวิจัย

การดำเนินงานวิจัยเพื่อพัฒนาเครื่องมือวิเคราะห์สมรรถนะพี-สแป เพื่อให้บรรลุตามวัตถุประสงค์นั้น เริ่มต้นจากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องดังรายละเอียดที่ได้กล่าวถึงในบทที่ 2 และ 3 ในบทที่ 4 นี้อธิบายถึงวิธีการดำเนินงานวิจัย ซึ่งมีห้าขั้นตอนหลัก ได้แก่ *ขั้นตอนแรก* การศึกษาความเป็นไปได้ ประกอบด้วย การคัดเลือกระบบจำลอง การศึกษาแนวทางเก็บข้อมูลจากระบบจำลอง และการกำหนดสภาพแวดล้อมที่ใช้ในการทำวิจัย *ขั้นตอนที่สอง* การพัฒนาโมดูลพี-สแปบนระบบจำลองซิมิกส์ เพื่อให้สามารถวิเคราะห์คุณสมบัติของพื้นที่เก็บข้อมูล จำแนกชนิดการไม่พบข้อมูลในแคช และนำเสนอข้อมูลการใช้ประโยชน์พื้นที่แคชในรูปแบบของกราฟได้ *ขั้นตอนที่สาม* การทดสอบความถูกต้องของพี-สแปเปรียบเทียบกับผลการประมวลผลบนระบบจริง *ขั้นตอนที่สี่* การออกแบบการทดลองเพื่อแสดงถึงศักยภาพของพี-สแปในการวิเคราะห์สมรรถนะของโปรแกรมคำนวณทางวิทยาศาสตร์ที่พัฒนาด้วยโอเพนเอ็มพี ซึ่งจะกล่าวถึงในบทที่ 5 และ 6 ของวิทยานิพนธ์ฉบับนี้ และ *ขั้นตอนสุดท้าย* คือการนำเสนอพี-สแป สรุปวิทยานิพนธ์ฉบับนี้ และข้อเสนอแนะของแนวทางการวิจัยในอนาคต ซึ่งจะกล่าวถึงในบทที่ 7

ในบทถัดไปของรายงานฉบับนี้ จะนำเสนอการทดลองเบื้องต้น ซึ่งเป็นการทดลองระหว่างการพัฒนาพี-สแปเพื่อศึกษาการทำงานของระบบจำลองซิมิกส์ ซึ่งแบ่งออกเป็นสามการทดลอง คือ การทดลองเพื่อคัดเลือกคำสั่งการแจกตัวนับดูป บนชุดโปรแกรมเอ็นพีบี 2.3 การทดลองเพื่อเปรียบเทียบสมรรถนะแคชระดับที่ 2 บนชิปมัลติโพรเซสเซอร์ และการวิเคราะห์คุณสมบัติการใช้พื้นที่เก็บข้อมูลของฟังก์ชันการคูณเมทริกซ์แบบขนาน