



การออกแบบวิธีการจัดรูปแบบข้อมูลเพื่อใช้ในการแก้ปัญหาระบบสมการเชิงเส้น
ด้วยวิธีการกำจัดของเกาส์บนระบบคลัสเตอร์

สมหมาย ศรีบัวบาล

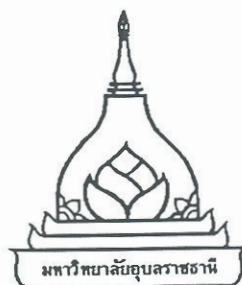
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

พ.ศ. 2548

ISBN 974-523-030-8

ลิขสิทธิ์เป็นของมหาวิทยาลัยอุบลราชธานี



**A NEW DATA LAYOUT FOR SOLVING LINEAR EQUATION SYSTEM
USING GAUSSIAN ELIMINATION ON COMPUTATIONAL CLUSTER**

SOMMAI SRIBUABAN

**A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
MAJOR IN INFORMATION TECHNOLOGY
FACULTY OF SCIENCE
UBON RAJATHANEE UNIVERSITY**

YEAR 2005

ISBN 974-523-030-8

COPYRIGHT OF UBON RAJATHANEE UNIVERSITY



ใบรับรองวิทยานิพนธ์
มหาวิทยาลัยอุบลราชธานี
ปริญญา วิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

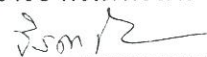
เรื่อง การออกแบบวิธีการจัดรูปแบบข้อมูลเพื่อใช้ในการแก้ปัญหาระบบสมการเชิงเส้น
ด้วยวิธีการกำจัดของเกาส์บนระบบคลัสเตอร์

ผู้วิจัย นายสมหมาย ศรีบัวบาล

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์

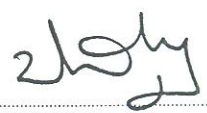

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อุทิศ อินทร์ประสิทธิ์)


..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ภัทรสินี ภัทรโกศล)


..... กรรมการ
(ดร.จิรดา กันทรารักษ์)


..... คณบดี
(ผู้ช่วยศาสตราจารย์ วรรณวิไล อธิวาสน์พงศ์)

มหาวิทยาลัยอุบลราชธานี รับรองแล้ว


.....
(ศาสตราจารย์ ดร.ประกอบ วิโรจน์กัญ)

อธิการบดี มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2548

กิตติกรรมประกาศ

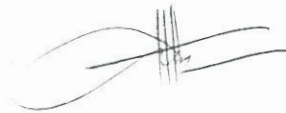
การวิจัยครั้งนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ดร.จิรดา กันทรารักษ์ ที่ช่วยเหลือให้คำแนะนำและข้อคิดเห็นในการวิจัยเป็นอย่างดีมาตลอดระยะเวลาที่ทำการวิจัย

ขอขอบคุณ ผศ.ดร. อุทิศ อินทร์ประสิทธิ์ รองคณบดีฝ่ายเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ที่ให้คำปรึกษา ชี้แนะในการศึกษาตลอดระยะเวลาการศึกษา ในมหาวิทยาลัยอุบลราชธานี

ขอขอบคุณ ผศ.ดร.มนูญ ศรีวิรัตน์ ประธานหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ รวมทั้งคณาจารย์ในภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ทุกท่าน ที่ให้คำปรึกษา ชี้แนะในการศึกษาตลอดระยะเวลาการศึกษาในมหาวิทยาลัยอุบลราชธานี

ขอขอบคุณสำนักงานคณะกรรมการอุดมศึกษาแห่งชาติและคณะบริหารศาสตร์ มหาวิทยาลัยอุบลราชธานีที่มอบทุนสนับสนุนการทำวิทยานิพนธ์ในครั้งนี้

ขอขอบคุณบิดา มารดา ครอบครัวและเพื่อน ๆ ที่ให้กำลังใจในการศึกษาและการทำงาน ตลอดมา จึงทำให้การวิจัยนี้สำเร็จลงด้วยดี



(นายสมหมาย ศรีบัวบาล)

ผู้วิจัย

บทคัดย่อ

ชื่อเรื่อง : การออกแบบวิธีการจัดรูปแบบข้อมูลเพื่อใช้ในการแก้ปัญหาระบบสมการเชิงเส้น
ด้วยวิธีการกำจัดของเกาส์บนระบบคลัสเตอร์

โดย : สมหมาย ศรีบัวบาล

ชื่อปริญญา : ปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชา : เทคโนโลยีสารสนเทศ [ISBN 974-523-030-8]

ประธานกรรมการที่ปรึกษา : ดร.จิรดา กันทรารักษ์

ศัพท์สำคัญ : วิธีการจัดรูปแบบข้อมูล วิธีการกำจัดของเกาส์ การประมวลผลแบบขนาน

วิทยานิพนธ์นี้เสนอวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ซึ่งเป็นรูปแบบวิธีการกระจายข้อมูลแบบใหม่ที่ออกแบบขึ้นเพื่อใช้ในการแก้ปัญหาสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์บนระบบคลัสเตอร์โดยอาศัยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกเป็นต้นแบบในการออกแบบ เนื่องจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกมีปัญหาในเรื่องของการไหลคบาลานซ์และการใช้ทรัพยากรได้ไม่คุ้มค่ามากนัก ส่วนวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกซึ่งพัฒนาต่อมาเพื่อแก้ปัญหาดังกล่าวแต่ก็มีปริมาณการสื่อสารเกิดขึ้นมาก ทำให้ไม่สามารถทำงานได้อย่างมีประสิทธิภาพเท่าที่ควร จึงได้ทำการปรับปรุงรูปแบบการกระจายข้อมูล เพื่อต้องการที่จะลดการไหลคบาลานซ์ในการทำงานแต่ไม่ทำให้เกิดการสื่อสารเพิ่มขึ้นมากนัก ซึ่งน่าจะทำได้ประสิทธิภาพมากยิ่งขึ้น หลังจากออกแบบวิธีการจัดรูปแบบข้อมูลแล้วได้พัฒนาโปรแกรมขึ้นมาทดสอบและทำการเปรียบเทียบเพื่อหาประสิทธิภาพในการทำงาน จากผลการทดสอบพบว่า วิธีการใหม่เป็นรูปแบบวิธีการที่มีประสิทธิภาพมากกว่าวิธีการต้นแบบ เพราะใช้เวลาในการทำงานน้อยกว่าและมีค่าอัตราความเร็วในการทำงานที่มากกว่า

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญตาราง	ง
สารบัญภาพ	ช
บทที่	
1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	3
1.3 ขอบเขตของการวิจัย	3
1.4 เครื่องมือที่ใช้ในการวิจัย	3
1.5 สถานที่และระยะเวลาการทำวิจัย	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ	4
1.7 ขั้นตอนในการดำเนินการวิจัย	4
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	
2.1 ทฤษฎีที่เกี่ยวข้อง	5
2.2 งานวิจัยที่เกี่ยวข้อง	21
3 การวิเคราะห์และออกแบบ	
3.1 การประมวลผลแบบลำดับ	23
3.2 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก	24
3.3 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก	25
3.4 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก	27
3.5 การวิเคราะห์เวลาในการทำงาน	28
4 การทดลองและผลการทดลอง	
4.1 การทดลอง	37
4.2 ผลการทดลอง	40
4.3 การเปรียบเทียบประสิทธิภาพในการทำงาน	48

สารบัญ (ต่อ)

	หน้า
5 สรุปผล	
5.1 สรุปผลการศึกษา	59
5.2 ข้อเสนอแนะ	61
เอกสารอ้างอิง	62
ภาคผนวก	65
ภาคผนวก ก การหาผลเฉลี่ยของระบบสมการเชิงเส้นด้วยวิธีกำจัดของเกาส์	66
ประวัติผู้วิจัย	72

สารบัญตาราง

ตารางที่		หน้า
1.1	แผนการดำเนินการวิจัย	3
3.1	ปริมาณการคำนวณของวิธีการทำงานแบบลำดับ	29
3.2	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	30
3.3	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 4	31
3.4	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 8	32
3.5	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 16	32
3.6	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 32	33
3.7	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 64	33
3.8	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 128	33
3.9	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 256	34
3.10	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกไขว้คลิก บล็อกข้อมูลขนาด 512	34
3.11	ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บล็อกและคอลัมน์ไขว้คลิก	35
4.1	ชุดข้อมูลที่ใช้ในการทดลอง	38
4.2	เวลาที่ใช้ในการประมวลผลแบบลำดับของวิธีการกำจัดของแก๊สแบบไม่ สลับแถว	40

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
4.3	เวลาที่ใช้ในการประมวลผลแบบลำดับของวิธีการกำจัดของเกาส์แบบ สลับแถว	41
4.4	เวลาที่ใช้ในการประมวลผลด้วยวิธีการกำจัดของเกาส์แบบ ไม่สลับแถว โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บนระบบคลัสเตอร์ ขนาด 4 หน่วยประมวลผล	42
4.5	เวลาที่ใช้ในการประมวลผลด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บนระบบคลัสเตอร์ขนาด 4 โหนด	43
4.6	เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่ สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บนบล็อกไฮคลิค ทำงาน บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	44
4.7	เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับ แถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บนบล็อกไฮคลิคบนระบบ คลัสเตอร์ขนาด 4 หน่วยประมวลผล	45
4.8	เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่ สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บนบล็อกและคอลัมน์ ไฮคลิคบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	46
4.9	เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับ แถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บนบล็อกและคอลัมน์ไฮคลิค บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	47
4.10	การเปรียบเทียบเวลาในการประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์ ระหว่างแบบไม่สลับแถวและแบบสลับแถว	48
4.11	การเปรียบเทียบเวลาในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บนบล็อกระหว่างแบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล	50
4.12	การเปรียบเทียบเวลาในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ บนบล็อกไฮคลิคระหว่างแบบที่ไม่มีการสลับแถวข้อมูลและแบบที่มีการสลับ แถวข้อมูล บนบล็อกข้อมูลขนาด 512	52

สารบัญตาราง (ต่อ)

หน้า

ตารางที่

- | | | |
|------|--|----|
| 4.13 | การเปรียบเทียบเวลาในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์
บล็อกและคอลัมน์ไขว้กันระหว่างแบบไม่มีการสลับแถวและแบบมีการสลับ
แถว | 53 |
| 4.14 | เปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นในการทำงานของแต่ละ
วิธีแบบมีการสลับแถวข้อมูล | 55 |
| 4.15 | เปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นในการทำงานของแต่ละ
วิธีแบบไม่มีการสลับแถวข้อมูล | 57 |

สารบัญภาพ

ภาพที่		หน้า
2.1	รูปแบบระบบสมการเชิงเส้นที่อยู่ในรูปแบบของ $Ax=b$ ที่มี n สมการ n ตัวแปร	5
2.2	รูปย่อของระบบสมการเชิงเส้น $Ax=b$ ที่มี n สมการ n ตัวแปร	6
2.3	รูปแบบของเมตริกซ์แต่งเต็ม $[A b]$ เพื่อใช้ในวิธีการกำจัดของเกาส์	6
2.4	การกำจัดตัวแปรที่ไม่ทราบค่าขั้นตอนที่ 1	7
2.5	การกำจัดตัวแปรที่ไม่ทราบค่าขั้นตอนที่ 2	7
2.6	การกำจัดตัวแปรที่ไม่ทราบค่าขั้นตอนที่ n	7
2.7	ขั้นตอนการส่งค่าที่ใช้สำหรับหารให้กับแต่ละแถว	9
2.8	ขั้นตอนการหารของการแก้สมการในแต่ละแถว	9
2.9	ขั้นตอนการส่งค่าให้กับแถวและคอลัมน์ที่อยู่ถัดไป	9
2.10	ขั้นตอนการคำนวณเพื่อกำจัดค่าสัมประสิทธิ์ของสมการ	10
2.11	ค่าสัมประสิทธิ์ของตัวแปรของสมการในรูปสามเหลี่ยมด้านล่างหลังจากเสร็จสิ้นขั้นตอนการกำจัดตัวแปรที่ไม่ทราบค่า	10
2.12	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก	13
2.13	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก	13
2.14	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก	14
2.15	วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อก	15
2.16	วิธีการจัดรูปแบบข้อมูลแบบแถวไซคลิก	15
2.17	วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อกไซคลิก	16
2.18	วิธีการจัดรูปแบบข้อมูลแบบแถวและคอลัมน์บล็อกไซคลิก	17
2.19	วิธีการจัดรูปแบบข้อมูลแบบสควเอบล็อก	17
2.20	ระบบคลัสเตอร์ที่มีขนาด 4 หน่วยประมวลผลเชื่อมโยงผ่านเครือข่ายความเร็วสูง	19
3.1	การกระจายข้อมูลแบบลำดับ เมตริกซ์ A ขนาด 8×8 เมตริกซ์ B ขนาด 8×1	23
3.2	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	24

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3.3	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ที่มี 4 หน่วยประมวลผล	25
3.4	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไชคลิกบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	26
3.5	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์และคอลัมน์ไชคลิก เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ 4 หน่วยประมวลผล	26
3.6	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไชคลิกบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล	27
3.7	วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์และคอลัมน์ไชคลิก เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ 4 หน่วยประมวลผล	28
4.1	กราฟแสดงการเปรียบเทียบเวลาในการประมวลผลแบบลำดับด้วยวิธีการกำจัดของแก๊สระหว่างแบบไม่สลับแถวและแบบสลับแถว	49
4.2	กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลแบบลำดับเมตริกซ์ขนาด 512×512 ถึง 4069×4096 ทั้งแบบมีการสลับแถวข้อมูลและแบบไม่มีการสลับแถว	49
4.3	กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกระหว่างแบบมีการสลับแถวข้อมูลและไม่มีการสลับแถวข้อมูล	51
4.4	กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ระหว่างแบบมีการสลับแถวและแบบไม่มีการสลับแถว	51
4.5	กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไชคลิกระหว่างแบบมีการสลับแถวข้อมูลและไม่มีการสลับแถวข้อมูล	52
4.6	กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไชคลิกระหว่างแบบมีการสลับแถวและไม่มีการสลับแถว	54

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
4.7	กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างแบบสลับแถวและแบบไม่มีการสลับแถว	54
4.8	กราฟแสดงการเปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก และวิธีการแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ทำงานแบบมีการสลับแถว	56
4.9	กราฟแสดงการเปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ทำงานแบบไม่มีการสลับแถว	58

บทที่ 1

บทนำ

ในบทนี้ จะกล่าวถึง ความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย สถานที่และระยะเวลาในการทำวิจัย ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย และ ขั้นตอนในการดำเนินการวิจัย

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการประมวลผลงานที่มีข้อมูลปริมาณมาก ๆ หรืองานที่มีขั้นตอนในการทำงานที่ซับซ้อน อาจจะต้องอาศัยเครื่องคอมพิวเตอร์ที่มีศักยภาพสูงในการประมวลผล ซึ่งโดยส่วนใหญ่ผู้ใช้จะนึกถึง ซูเปอร์คอมพิวเตอร์ ซึ่งเป็นคอมพิวเตอร์ที่มีประสิทธิภาพในการทำงานสูง สามารถรองรับการทำงานที่มีปริมาณมาก ๆ และมีความซับซ้อนได้ แต่เนื่องจากซูเปอร์คอมพิวเตอร์มีราคาค่อนข้างสูง จึงทำให้องค์กรต่าง ๆ ไม่สามารถจัดสรรงบประมาณเพื่อซื้อเครื่องซูเปอร์คอมพิวเตอร์มาใช้งานได้ จึงเกิดแนวความคิด [1] ที่จะนำเอาเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) มาต่อพ่วงกัน หลาย ๆ ตัวเป็นระบบเครือข่ายคอมพิวเตอร์ เพื่อใช้ประโยชน์จากซีพียูของคอมพิวเตอร์แต่ละเครื่อง ด้วยการแบ่งงานให้ซีพียูแต่ละตัวช่วยกันประมวลผล ทำให้สามารถคำนวณงานที่มีขนาดใหญ่ได้เร็วขึ้นโดยไม่จำเป็นต้องใช้ซูเปอร์คอมพิวเตอร์ ระบบที่สร้างขึ้นจากการเชื่อมต่อซีพียูนี้เรียกว่า ระบบคลัสเตอร์ (Cluster Computer) ซึ่งการทำงานของระบบคลัสเตอร์นั้นเป็นการประมวลผลแบบขนาน (Parallel Computing) [2] เพราะฉะนั้น โปรแกรมที่จะสามารถทำงานบนคลัสเตอร์คอมพิวเตอร์ได้อย่างมีประสิทธิภาพนั้น ก็จะต้องรองรับการทำงานแบบขนานด้วยเช่นกัน

เนื่องจากการประมวลผลแบบขนานนั้นทำงานบนระบบคลัสเตอร์ที่เกิดจากการเชื่อมโยงคอมพิวเตอร์หลาย ๆ เครื่องเข้าด้วยกันเพื่อช่วยกันประมวลผลการทำงาน เพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพจึงจำเป็นต้องหาวิธีการที่ใช้แก้ปัญหาที่มีขั้นตอนการทำงานแบบมีอิสระ เมื่อพิจารณาวิธีการที่ใช้ในการแก้ปัญหาทางคณิตศาสตร์พบว่า วิธีการกำจัดของเกาส์เป็นหนึ่งในวิธีที่นิยมใช้ในการแก้ปัญหาระบบสมการเชิงเส้น เพราะเป็นวิธีการที่ใช้หาผลเฉลยของระบบสมการเชิงเส้นที่ไม่ซับซ้อนมาก และมีขั้นตอนในการทำงานที่มีอิสระ เมื่อนำมาทำงานบนระบบขนานน่าจะสามารถทำงานได้อย่างมีประสิทธิภาพ

สิ่งที่มีผลต่อประสิทธิภาพในการทำงานเมื่อนำเอาวิธีการแก้ปัญหามาทางคณิตศาสตร์มาทำงานแบบขนานอีกอย่างหนึ่งก็คือ วิธีการจัดรูปแบบข้อมูล (Data Layout) ซึ่งแต่ละวิธีการจะมีการสื่อสาร (Communication) และการคำนวณ (Computation) เกิดขึ้นในปริมาณที่แตกต่างกันไป โดยทั้งสองตัวนี้เป็นดัชนีชี้วัดประสิทธิภาพที่สำคัญ หากจะนำเอาวิธีการจัดรูปแบบข้อมูลมาทำงานโดยใช้วิธีการกำจัดของเกาส์จะต้องหาวิธีการจัดรูปแบบที่เหมาะสมกับวิธีการนี้ เนื่องจากกระบวนการในการกำจัดค่าตัวแปรด้วยวิธีการกำจัดของเกาส์นั้นมีขั้นตอนของการสลับแถวข้อมูลเพื่อลดข้อผิดพลาดจากค่าในแนวเส้นทแยงมุมที่นำมาหารเป็น 0 การสลับแถวนั้นจะทำการสลับแถวที่อยู่ในคอลัมน์เดียวกัน จึงจำเป็นต้องใช้วิธีการจัดรูปแบบข้อมูลที่มีลักษณะการทำงานแบบคอลัมน์ (Column) เพราะว่า ถ้าหากใช้วิธีการจัดรูปแบบข้อมูลแบบอื่นจะทำให้เกิดการสื่อสารเกิดขึ้นในปริมาณที่มาก ซึ่งมีผลต่อประสิทธิภาพในการทำงาน สำหรับวิธีการจัดรูปแบบข้อมูลที่มีลักษณะการทำงานแบบคอลัมน์นั้นมีหลายวิธีด้วยกันคือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก (Column Block), วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก (Column Cyclic), วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก (Column Block Cyclic) สำหรับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกนั้นถือว่าเป็นต้นแบบของการทำงานแบบคอลัมน์ที่มีขั้นตอนการทำงานที่ง่ายแต่ก็มีข้อเสียในเรื่องความคุ้มค่าในการใช้ทรัพยากรและมีการคำนวณในปริมาณที่มาก ส่วนวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกเป็นวิธีที่ปรับปรุงมาจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ซึ่งทำให้ใช้ทรัพยากรในการทำงานได้คุ้มค่าขึ้นและสามารถลดปริมาณการคำนวณลงได้เล็กน้อย แต่วิธีการทั้งสองก็มีปริมาณการสื่อสารเกิดขึ้นมากพอสมควรจึงไม่ได้ประสิทธิภาพเพิ่มมากขึ้นเท่าที่ควร ดังนั้นหากมีวิธีการจัดรูปแบบข้อมูลที่มีปริมาณการสื่อสารใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและมีปริมาณการคำนวณใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกน่าจะทำงานได้ดีกว่าทั้งสามวิธี

ในงานวิจัยนี้ จะเป็นการออกแบบวิธีการจัดรูปแบบข้อมูลสำหรับการแก้ปัญหาระบบสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์เพื่อใช้ในการกระจายข้อมูล (Scatter Data) แบบขนานบนระบบเครือข่ายคลัสเตอร์ จากนั้นจะทำการเปรียบเทียบประสิทธิภาพ ในเรื่องของการใช้เวลาในการทำงาน (Execution Time) และอัตราความเร็วในการทำงานที่เพิ่มขึ้น (Speedup) ของวิธีการจัดรูปแบบข้อมูลที่สร้างขึ้นใหม่เพื่อเปรียบเทียบกับการประมวลผลแบบลำดับ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก

1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบวิธีการจัดรูปแบบข้อมูลสำหรับวิธีการกำจัดของเกาส์ที่ใช้ในการกระจายข้อมูลแบบขนานบนระบบเครือข่ายคลัสเตอร์ และเปรียบเทียบให้เห็นถึงประสิทธิภาพความเร็วในการประมวลผลเมื่อเทียบกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก

1.3 ขอบเขตของการวิจัย

สำหรับงานวิจัยชิ้นนี้ เป็นการออกแบบที่ให้ความสำคัญในเรื่องของการออกแบบวิธีการในการแจกจ่ายการทำงานให้แก่คอมพิวเตอร์แต่ละตัวตามวิธีการจัดรูปแบบข้อมูลที่ออกแบบไว้ จะไม่เน้นที่ตัวฮาร์ดแวร์ (Hardware) หรืออุปกรณ์ที่จะติดต่อกันๆ โดยทำการออกแบบวิธีการจัดรูปแบบข้อมูลและเขียนโปรแกรมเพื่อให้สามารถทำงานได้บนระบบขนานตามที่ต้องการ นั่นก็คือ ให้สามารถแบ่งการทำงานในการช่วยกันประมวลผลของวิธีการกำจัดของเกาส์ได้อย่างรวดเร็วยิ่งขึ้น นอกจากนี้ยังทำการรวมผลสรุปการทำงาน โดยเปรียบเทียบเวลาที่ใช้ในการทำงานและอัตราความเร็วในการทำงานที่เพิ่มขึ้นเพื่อให้เห็นประสิทธิภาพที่แตกต่างกันในแต่ละรูปแบบ

1.4 เครื่องมือที่ใช้ในการวิจัย

1.4.1 ฮาร์ดแวร์ระบบประกอบด้วย

ระบบคลัสเตอร์คอมพิวเตอร์ ประกอบไปด้วย

- 1) เครื่องคอมพิวเตอร์พีซี จำนวน 5 เครื่อง
- 2) การ์ดเน็ตเวิร์คจำนวน 6 ใบ
- 3) เน็ตเวิร์คสวิตช์จำนวน 1 ตัว

1.4.2 ซอฟต์แวร์ระบบ

- 1) ระบบปฏิบัติการลินุกซ์ที่ใช้เคอร์เนลตั้งแต่เวอร์ชัน 2.4.x ขึ้นไป
- 2) คอมไพเลอร์ภาษาซี/ซีพลัสพลัสที่ทำงานบนระบบปฏิบัติการลินุกซ์
- 3) ซอฟต์แวร์สำหรับการประมวลผลแบบขนาน MPI

1.5 สถานที่และระยะเวลาการวิจัย

สถานที่ที่ใช้ในการทำวิทยานิพนธ์นี้คือ ห้องปฏิบัติการเครือข่ายคอมพิวเตอร์คลัสเตอร์ ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ มหาวิทยาลัยอุบลราชธานี

ระยะเวลาในการทำงานวิจัย จะเริ่มต้นการทำงานตั้งแต่เดือนพฤศจิกายน พ.ศ. 2547 จนถึงเดือนมีนาคม พ.ศ. 2548 จำนวน 20 สัปดาห์ ดังแสดงในตารางที่ 1.1

ตารางที่ 1.1 แผนการดำเนินการวิจัย

ลำดับ	กิจกรรม	สัปดาห์ที่ (จำนวน 20 สัปดาห์)									
		1-2	3-4	5-6	7-8	9-10	11-12	13-14	15-16	17-18	19-20
1	ศึกษาและออกแบบวิธีการจัดรูปแบบข้อมูล										
2	เขียนโปรแกรมตามรูปแบบที่ออกแบบไว้										
3	ทดสอบประสิทธิภาพ										
4	เปรียบเทียบการทำงาน										
5	ปรับปรุงข้อผิดพลาดในการทำงาน										
6	เขียนรายงานวิจัย										

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถช่วยในการหาผลลัพธ์ของสมการให้เร็วขึ้น
- 2) สามารถนำไปประยุกต์ใช้กับการแก้ปัญหาอื่น ๆ ได้
- 3) สามารถพัฒนาเป็นซอฟต์แวร์เชิงพาณิชย์เพื่อลดการนำเข้า

1.7 ขั้นตอนในการดำเนินการวิจัย

- 1) ศึกษาการแก้ระบบสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์
- 2) ศึกษาวิธีการทำงานแบบขนานบนระบบคลัสเตอร์ และวิธีการจัดรูปแบบข้อมูลต่าง ๆ ให้แก่เครื่องคอมพิวเตอร์ที่อยู่ในระบบเครือข่ายคลัสเตอร์ โดยใช้ MPI ซึ่งเป็นโปรแกรมที่ทำหน้าที่เป็น Interface ที่โปรแกรมใช้ในการติดต่อสื่อสารกับเครื่องอื่น ๆ ภายใต้อุปกรณ์เครือข่ายคลัสเตอร์
- 3) ออกแบบวิธีการจัดรูปแบบข้อมูลสำหรับวิธีการกำจัดของเกาส์บนระบบขนาน
- 4) เขียนโปรแกรมโดยใช้ MPI เพื่อให้ทำงานตามวิธีการจัดรูปแบบข้อมูลที่ออกแบบไว้
- 5) ทดสอบการทำงานของวิธีการจัดรูปแบบข้อมูลที่สร้างขึ้น ทำการเปรียบเทียบการทำงานในเครือข่ายคลัสเตอร์โดยเปรียบเทียบระยะเวลาของการทำงานเมื่อทำงานแบบลำดับกับการทำงานแบบขนาน
- 6) ปรับปรุงข้อผิดพลาดที่เกิดจากการทำงาน
- 7) จัดทำรายงานการวิจัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้เป็นการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง ประกอบด้วย 2 ส่วน ส่วนที่ 1 ทฤษฎีที่เกี่ยวข้อง ได้นำเสนอ วิธีการกำจัดของเกาส์ (Gaussian Elimination), การสลับแถวเพื่อลดข้อผิดพลาด (Pivoting), การกระจายข้อมูล (Scatter Data), ขั้นตอนวิธี (Algorithm), ระบบคลัสเตอร์ (Cluster Computer), การประมวลผลแบบขนาน (Parallel Computing) และ MPI (Message Passing Interface) ส่วนที่ 2 งานวิจัยที่เกี่ยวข้องได้นำเสนอ งานวิจัยต่าง ๆ ที่มีนักวิจัยได้ทำการศึกษา ซึ่งเกี่ยวข้องกับวิธีการจัดรูปแบบข้อมูลเพื่อที่จะนำมาใช้กระจายข้อมูลในการแก้ปัญหาสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 วิธีการกำจัดของเกาส์ (Gaussian Elimination)

วิธีการกำจัดของเกาส์ [3] เป็นวิธีที่ใช้การหาผลเฉลยของระบบสมการเชิงเส้น โดยการจัดรูปสมการใหม่เพื่อกำจัดตัวไม่ทราบค่าในสมการเชิงเส้น ซึ่งเป็นการทำให้ค่าสัมประสิทธิ์ของตัวแปรที่อยู่ด้านซ้ายของแนวเส้นทแยงมุมเป็นศูนย์โดยใช้วิธีการลดทอน วิธีนี้เป็นวิธีที่มีการใช้มานาน ตั้งแต่ยุคแรก ๆ ของการคำนวณและยังเป็นที่นิยมใช้อยู่ในปัจจุบัน อีกทั้งยังเป็นที่มาของวิธีการแก้ปัญหาในโปรแกรมสำเร็จรูปหลาย ๆ โปรแกรมด้วย โดยผู้คิดค้นวิธีนี้ขึ้นมาคือ คาร์ล ฟรีดริช เกาส์ (Carl Friedrich Gauss) [4] นักคณิตศาสตร์ชาวเยอรมัน รูปแบบของสมการซึ่งโดยทั่วไปแล้วระบบสมการที่มี n สมการ n ตัวแปร จะอยู่ในรูปของ $Ax=b$ ดังนี้

$$\begin{pmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots + \vdots + \ddots + \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

ภาพที่ 2.1 รูปแบบระบบสมการเชิงเส้นที่อยู่ในรูปแบบของ $Ax=b$ ที่มี n สมการ n ตัวแปร

ซึ่งสามารถที่จะเขียนรูปแบบของสมการแบบย่อได้ดังนี้

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

ภาพที่ 2.2 รุปย่อของระบบสมการเชิงเส้น $Ax=b$ ที่มี n สมการ n ตัวแปร [3]

โดยทั่วไปแล้วการแก้ปัญหสมการเชิงเส้นโดยใช้วิธีการกำจัดของเกาส์นั้นจะมีการสร้างเมตริกซ์ของสมการที่ต้องการแก้ไขอยู่ในรูปแบบของเมตริกซ์แต่งเติม $[A \mid b]$ ดังนี้

$$\left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array} \right)$$

ภาพที่ 2.3 รูปแบบของเมตริกซ์แต่งเติม $[A \mid b]$ เพื่อใช้ในวิธีการกำจัดของเกาส์ [3]

สำหรับการแก้ปัญหสมการเชิงเส้นโดยใช้วิธีการกำจัดของเกาส์ประกอบด้วย 2 ขั้นตอนดังนี้ [5]

- 1) การกำจัดตัวแปรที่ไม่ทราบค่า (Forward Elimination) เป็นขบวนการกำจัดตัวแปรที่ไม่ทราบค่าทั้งหมดที่มีอยู่ในสมการเชิงเส้น
- 2) การแทนค่าย้อนกลับ (Back-substitutions) เป็นการหาค่าของตัวแปรโดยใช้การแทนค่าย้อนกลับ

การกำจัดตัวแปรที่ไม่ทราบค่าด้วยวิธีการกำจัดของเกาส์มีขั้นตอนต่าง ๆ สำหรับระบบสมการ n สมการ n ตัวแปร $Ax=b$ ดังนี้ [6]

- 1) สร้างเมตริกซ์แต่งเติมให้อยู่ในรูปแบบของ $[A \mid b]$
- 2) ทำการสลับแถว (ถ้าจำเป็น) เพื่อให้ได้เมตริกซ์ที่มีคุณสมบัติว่าสมาชิกที่ตำแหน่งแถวที่ i หลักที่ i มีค่าสัมบูรณ์มากที่สุดเป็นหลักที่ i นั้น และถ้าค่านั้นไม่เท่ากับศูนย์เราจะคำนวณต่อไป
- 3) ทำการดำเนินการเปลี่ยนแถวกับเมตริกซ์ที่ได้ เพื่อให้สมาชิกในหลักที่ i ของแถวที่ $2, 3, 4, \dots, n$ เป็นศูนย์ทุกตัวได้เมตริกซ์ $[A \mid b]^{(i)}$ ดังแสดงในภาพที่ 2.4

$$\left(\begin{array}{ccccc|c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & b_3^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right)$$

ภาพที่ 2.4 การกำจัดตัวแปรที่ไม่ทราบค่าขั้นตอนที่ 1

4) ะทำการเช่นเดียวกับขั้นตอนที่ 2 กับเมตริกซ์ที่ได้ในขั้นตอนที่ 3 โดยทำการสลับแถว (ถ้าจำเป็น) ระหว่างแถวที่ 2, 3, ..., n เพื่อให้ได้เมตริกซ์ที่มีคุณสมบัติว่าสมาชิกที่ตำแหน่งแถวที่ 2 หลักที่ 2 มีค่าสัมบูรณ์มากที่สุดเป็นหลักที่ 2 ถ้าค่านั้นไม่เป็นศูนย์เรากำหนดค่าไป

5) ทำการดำเนินการเปลี่ยนแถวกับเมตริกซ์ที่ได้ เพื่อให้สมาชิกในหลักที่ 2 ของแถวที่ 3, 4, ..., n เป็นศูนย์ทุกตัวได้เมตริกซ์ $[A:b]^{(2)}$ เป็น

$$\left(\begin{array}{ccccc|c} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} & b_1^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array} \right)$$

ภาพที่ 2.5 การกำจัดตัวแปรที่ไม่ทราบค่าขั้นตอนที่ 2

ในทำนองเดียวกันเมื่อทำต่อไปและ $a_{ii}^{(i)} \neq 0, i=1, 2, \dots, n$ เราจะได้เมตริกซ์ที่สมมูลแถวกับเมตริกซ์ของระบบสมการเป็น

$$\left(\begin{array}{ccccc|c} a_{11}^{(n)} & a_{12}^{(n)} & a_{13}^{(n)} & \cdots & a_{1n}^{(n)} & b_1^{(n)} \\ 0 & a_{22}^{(n)} & a_{23}^{(n)} & \cdots & a_{2n}^{(n)} & b_2^{(n)} \\ 0 & 0 & a_{33}^{(n)} & \cdots & a_{3n}^{(n)} & b_3^{(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n)} & b_n^{(n)} \end{array} \right)$$

ภาพที่ 2.6 การกำจัดตัวแปรที่ไม่ทราบค่าขั้นตอนที่ n

หลังจากเสร็จสิ้นกระบวนการกำจัดตัวแปรที่ไม่ทราบค่า ขั้นตอนต่อไปคือ การหาค่าของตัวแปร โดยใช้การแทนค่าย้อนกลับ ซึ่งหาได้จากสมการ [3, 7] ต่อไปนี้

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^n a_{ij} x_j \right) \quad (1)$$

สำหรับข้อมูลที่จะนำมาใช้กับวิธีการกำจัดของเกาส์ ส่วนใหญ่เป็นสมการเชิงเส้นที่อยู่ในรูปของเมตริกซ์ ซึ่งระบบสมการเชิงเส้นเป็นพื้นฐานสำคัญในเรื่องของการคำนวณที่มีการนำไปใช้งานอย่างแพร่หลาย เช่น การคำนวณทางฟิสิกส์ การพยากรณ์สภาพอากาศ และการวิเคราะห์ข้อมูลหุ้น

2.1.2 การสลับแถวของสมการเพื่อลดข้อผิดพลาด (Pivoting Row)

การใช้วิธีการกำจัดของเกาส์เพื่อหาผลเฉลยของสมการเชิงเส้น เป็นการทำให้ค่าสัมประสิทธิ์ของตัวแปรเป็นศูนย์โดยใช้การลดทอน ซึ่งวิธีการนี้ [8] จะทำโดยการเอาค่าสัมประสิทธิ์ในแนวเส้นทะแยงมุมของสมการ ไปเป็นตัวหาร หากมีสมาชิกของเมตริกซ์ในแนวเส้นทะแยงมุมตัวใดตัวหนึ่งมีค่าสัมประสิทธิ์เป็นศูนย์หรือมีค่าสัมประสิทธิ์เข้าใกล้ศูนย์มาก ๆ จะทำให้ผลที่ได้จากการหารมีข้อผิดพลาดหรือมีความคลาดเคลื่อน เพราะฉะนั้น [9, 10, 11] จึงใช้วิธีการสลับแถว เพื่อให้ได้สมาชิกของเมตริกซ์ที่มีค่าสัมประสิทธิ์สูงสุด ไปใช้เป็นตัวหารเพื่อเป็นการลดข้อผิดพลาดที่เกิดจากตัวหารมีค่าเป็นศูนย์ ในการสลับแถวข้อมูลนั้นจะทำการตรวจหาค่าสัมประสิทธิ์ของเมตริกซ์ที่มีค่าสมบูรณ์สูงสุดในคอลัมน์หลักที่กำลังทำงาน หากค่าสมบูรณ์สูงสุดไม่ได้อยู่ที่แถวหลักที่กำลังทำงานอยู่ จะทำการสลับแถวข้อมูลระหว่างแถวหลักที่กำลังทำงานอยู่ กับแถวที่มีค่าสมบูรณ์สูงสุด ซึ่งการสลับแถวของสมการจะเกิดปัญหาเรื่องของการเร็วในการทำงานที่ลดลงหากเปรียบเทียบกับ การหาค่าของสมการโดยการไม่สลับแถว เนื่องจากต้องใช้เวลาในการตรวจสอบหาค่าสมบูรณ์สูงสุดในแต่ละคอลัมน์และต้องใช้เวลาในการสลับแถว

2.1.3 ขั้นตอนวิธี (Algorithm) สำหรับใช้ในการแก้ปัญหาวิธีการกำจัดของเกาส์

2.1.3.1 ขั้นตอนวิธีพื้นฐานสำหรับวิธีการกำจัดของเกาส์

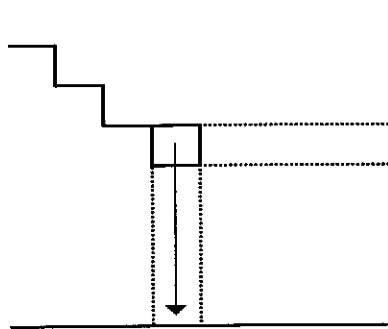
กระบวนการทำงานของวิธีการกำจัดของเกาส์ จะทำงานในแต่ละขั้นตอนเป็นลักษณะขั้นบันได โดยขั้นตอนวิธีพื้นฐานจะเป็นดังนี้

```

for i=0 to n-1 do
  pivoting step
  for j=i+1 to n do
    for k=i+1 to n+1 do
      A[j][k]=A[j][k]-(A[j][i]/A[i][i]*A[i][k]);
    end
  end
end
end

```

ในแต่ละขั้นตอนของการกำจัดตัวแปรที่เราไม่ทราบค่านั้นจะเริ่มจากการส่งค่า $A[i][i]$ ที่ใช้ในการหารให้กับแถวที่อยู่ด้านล่างของแถวกำลังทำงานอยู่ในปัจจุบัน ดังรูปภาพที่ 2.7



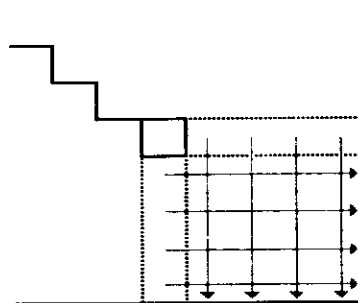
ภาพที่ 2.7 ขั้นตอนการส่งค่าที่ใช้สำหรับหารให้กับแต่ละแถว [5]

หลังจากแต่ละแถวได้รับค่า $A[i][i]$ แล้วจะทำการคำนวณโดยการหาร $A[j][i]/A[i][i]$ ดังรูปภาพที่ 2.8



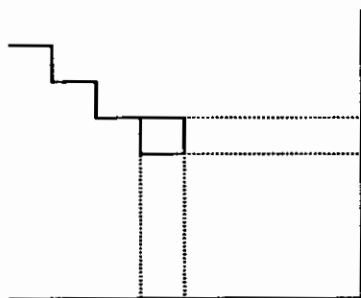
ภาพที่ 2.8 ขั้นตอนการหารของการแก้สมการในแต่ละแถว [5]

หลังจากทำการหารเรียบร้อยแล้วในแต่ละแถวจะส่งค่าที่ได้จากการหารให้กับคอลัมน์ถัดไปที่อยู่ในแถวเดียวกัน และในแต่ละคอลัมน์ของแถวที่ทำงานอยู่ในปัจจุบันจะส่งค่า $A[i][k]$ ให้แถวที่อยู่ถัดไป ดังรูปภาพที่ 2.9



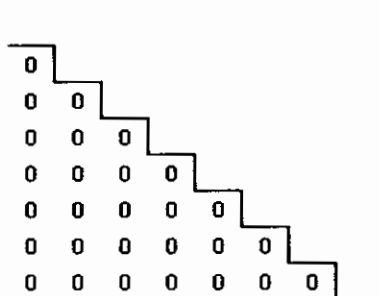
ภาพที่ 2.9 ขั้นตอนการส่งค่าให้กับแถวและคอลัมน์ที่อยู่ถัดไป [5]

หลังจากที่ได้รับค่าในแต่ละแถวจะทำการคำนวณเพื่อหาค่า $A[j][k]$ ซึ่งการทำงานในขั้นตอนนี้จะเกิดขึ้นในคอลัมน์หลักที่กำลังทำงานอยู่เป็นคืบไปจนถึงคอลัมน์สุดท้าย ดังแสดงในภาพที่ 2.10 โดยพื้นที่สีเทาคือพื้นที่ที่มีการคำนวณเพื่อกำจัดค่าสัมประสิทธิ์ของสมการ



ภาพที่ 2.10 ขั้นตอนการคำนวณเพื่อกำจัดค่าสัมประสิทธิ์ของสมการ [5]

ในการหาผลเฉลยของสมการจะทำตามขั้นตอนข้างต้นจนกว่าสามเหลี่ยมด้านล่างของสมการ (Lower triangular) มีค่าสัมประสิทธิ์เป็น 0 ทุกตัว ดังแสดงในภาพที่ 2.11



ภาพที่ 2.11 ค่าสัมประสิทธิ์ของตัวแปรของสมการในรูปสามเหลี่ยมด้านล่างหลังจากเสร็จสิ้นขั้นตอนการกำจัดตัวแปรที่ไม่ทราบค่า [5]

2.1.3.2 ขั้นตอนวิธีแบบเชิงแถว (Row oriented Algorithm)

ขั้นตอนวิธีแบบเชิงแถว [12, 13] เป็นขั้นตอนวิธีแบบหนึ่งที่นิยมใช้ในงานวิจัยต่าง ๆ โดยใช้ในการแก้ปัญหาวีธีการกำจัดของเกาส์ ขั้นตอนวิธีแบบเชิงแถวได้นำเอาวิธีการพื้นฐานมาปรับปรุงใหม่ เพื่อลดปริมาณการคำนวณลง ซึ่งการทำงานของขั้นตอนวิธีนี้ มีรูปแบบการทำงานอยู่ในลักษณะของแถว โดยจะแบ่งการคำนวณในวิธีการกำจัดของเกาส์ออกเป็น 2 ขั้นตอน คือ ขั้นตอนการหาร (Division Step) และขั้นตอนการคูณและการลบ (Multiplications and Subtractions Step) โดยทั้งสองขั้นตอนที่เกิดขึ้น จะทำงานให้เสร็จเป็นแถว ๆ เมื่อทำงานในแถวแรกเสร็จแล้ว จึงจะทำในแถวถัด ๆ ไป ขั้นตอนวิธีแบบเชิงแถวจะเป็นดังนี้

```

for i=0 to n-1 do
  pivot=i
  for j=i to n+1 do
    if |A[j][i]| ≥ |A[i][i]| then pivot=j
  end
  if pivot ≠ i then {
    for j=1 to n+1 do
      temp = A[i][j]
      A[i][j] = A[pivot][j]
      A[pivot][j] = temp
    end
  }
  for j=i+1 to n do
    A[j][i] = A[j][i] / A[i][i]
    for k=i+1 to n+1 do
      A[j][k] = A[j][k] - (A[j][i] * A[i][k])
    end
  end
end
end

```

2.1.3.3 ขั้นตอนวิธีแบบเชิงคอลัมน์ (Column oriented Algorithm)

ขั้นตอนวิธีแบบเชิงคอลัมน์ [12, 13] เป็นขั้นตอนวิธีแบบหนึ่งที่นิยมใช้ในงานวิจัยต่าง ๆ โดยใช้ในการแก้ปัญหาวิธีการกำจัดของเกาส์เหมือนขั้นตอนวิธีแบบเชิงแถว แต่มีรูปแบบการทำงานอยู่ในลักษณะของคอลัมน์ โดยจะแบ่งการคำนวณในวิธีการกำจัดของเกาส์ออกเป็น 2 ขั้นตอน คือ ขั้นตอนการหารและขั้นตอนการคูณและการลบโดยทั้งสองขั้นตอนที่เกิดขึ้น จะทำงานให้เสร็จเป็นคอลัมน์ ๆ เมื่อทำงานในคอลัมน์แรกเสร็จแล้ว จึงจะทำในคอลัมน์ถัด ๆ ไป ขั้นตอนวิธีแบบเชิงคอลัมน์จะเป็นดังนี้

```

for i=0 to n-1 do
  pivot=i
  for j=i to n+1 do
    if |A[j][i]| ≥ |A[i][i]| then pivot=j
  end
  if pivot ≠ i then {
    for j=1 to n+1 do
      temp = A[i][j]
      A[i][j] = A[pivot][j]
      A[pivot][j] = temp
    end
  }
  for j=i+1 to n do
    A[j][i] = A[j][i] / A[i][i]
  end
  for j=i+1 to n do
    for k=i+1 to n+1 do
      A[j][k] = A[j][k] - (A[j][i] * A[i][k])
    end
  end
end
end

```

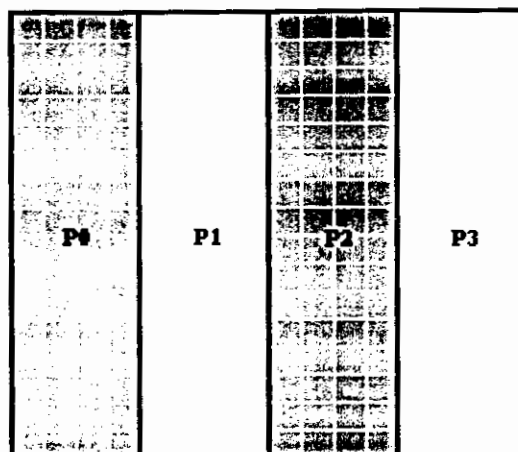

2.1.4 การกระจายข้อมูล (Scatter Data)

การกระจายข้อมูล (Scatter Data) [1] หรือเรียกอีกอย่างหนึ่งว่าวิธีการจัดรูปแบบข้อมูล (Data Layout) หมายถึง รูปแบบวิธีการที่ใช้แบ่งข้อมูลและกระจายข้อมูลไปยังโหนดต่าง ๆ ที่มีอยู่ในระบบคลัสเตอร์ โดยมีหน่วยประมวลผลหลัก (Source Node) เป็นตัวส่งข้อมูลไปยังหน่วยประมวลผลลูกข่าย (Cluster Node) ทุก ๆ ตัวที่อยู่ในระบบคลัสเตอร์เพื่อช่วยกันประมวลผลข้อมูล ระบบคลัสเตอร์เป็นการเชื่อมต่อเครื่องคอมพิวเตอร์แบบพีซี หลาย ๆ เครื่องบนระบบเครือข่ายความเร็วสูง โดยมีรูปแบบของโครงสร้างคอมพิวเตอร์แบบขนานที่มีหน่วยความจำแยกจากกันโดยเด็ดขาด เครื่องคอมพิวเตอร์แต่ละเครื่องจะมีหน่วยความจำสำหรับใช้งานเป็นของตัวเอง [2] เครื่องคอมพิวเตอร์แต่ละเครื่องจะติดต่อสื่อสารกันโดยใช้วิธีการแลกเปลี่ยนข้อความผ่านเครือข่ายคอมพิวเตอร์ ซึ่งจะมีการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์แต่ละเครื่องบนระบบคลัสเตอร์ที่ทำการประมวลผลข้อมูลอยู่ นั้น การส่งข้อมูล (Send) และการรับข้อมูล (Receive) ระหว่างหน่วยประมวลผลที่เกิดขึ้นในแต่ละครั้งจะมีผลต่อปริมาณของการสื่อสาร (Communication) โดย James Demmel ได้กล่าวเอาไว้ในงานวิจัยของเขาว่า [14] ประสิทธิภาพในการทำงานของโปรแกรมที่พัฒนาจะดีหรือไม่ดีนั้นขึ้นอยู่กับปริมาณการสื่อสารที่เกิดขึ้นในขณะที่มีการประมวลผล ดังนั้น [15] เพื่อที่จะทำให้การประมวลผลเป็นไปอย่างมีประสิทธิภาพจะต้องมีการออกแบบขั้นตอนวิธีแบบขนาน (Parallel Algorithm) ให้มีประสิทธิภาพด้วย สิ่งที่จะต้องคำนึงถึงก็คือวิธีการจัดรูปแบบข้อมูล การเลือกวิธีการจัดรูปแบบข้อมูลที่เหมาะสมกับประเภทของข้อมูลจะช่วยลดปัญหาต่าง ๆ ที่เป็นสาเหตุที่ทำให้เกิดการสื่อสาร และช่วยเพิ่มประสิทธิภาพในการทำงานได้มากขึ้น

จากการศึกษาค้นคว้าพบว่า มีงานวิจัยหลาย ๆ งานที่ให้ความสนใจในเรื่องของวิธีการจัดรูปแบบข้อมูล จึงทำให้เกิดวิธีการจัดรูปแบบข้อมูลที่หลากหลายเกิดขึ้น ซึ่งประสิทธิภาพของแต่ละรูปแบบข้อมูลก็แตกต่างกันไป สำหรับวิธีการจัดรูปแบบข้อมูลที่น่าสนใจนำมาใช้แก้ปัญหาในวิธีการกำจัดของเกาส์มีดังนี้

2.1.4.1 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก (Column Block)

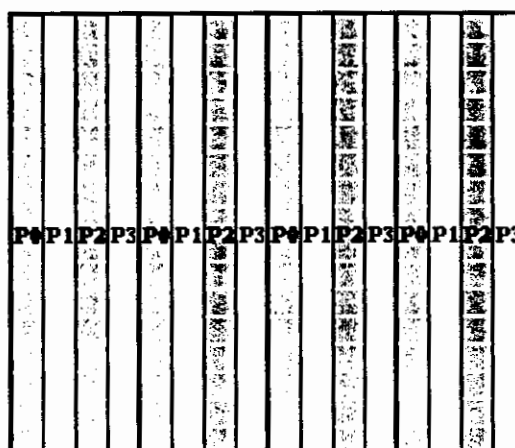
วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก [16] เป็นหนึ่งในวิธีการจัดรูปแบบข้อมูลที่น่าสนใจนำมาใช้เพื่อกระจายข้อมูลในการแก้ปัญหามหาสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์ โดยจะแบ่งข้อมูลออกเป็นบล็อกของคอลัมน์ที่มีขนาดเท่ากัน ในแต่ละบล็อกมีขนาดเท่ากับ n/p เมื่อ n หมายถึงขนาดของเมทริกซ์ และ p หมายถึง จำนวนหน่วยประมวลผลทั้งหมดที่มีในระบบคลัสเตอร์ ข้อมูลบล็อกแรกจะถูกแบ่งให้กับหน่วยประมวลผลตัวแรก ข้อมูลบล็อกถัดไปจะถูกแบ่งให้กับหน่วยประมวลผลตัวถัดไป โดยจะทำอย่างนี้ไปเรื่อย ๆ จนกว่าหน่วยประมวลผลทุกตัวจะได้รับข้อมูลจนครบ ดังแสดงในภาพที่ 2.12



ภาพที่ 2.12 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

2.1.4.2 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก (Column Cyclic)

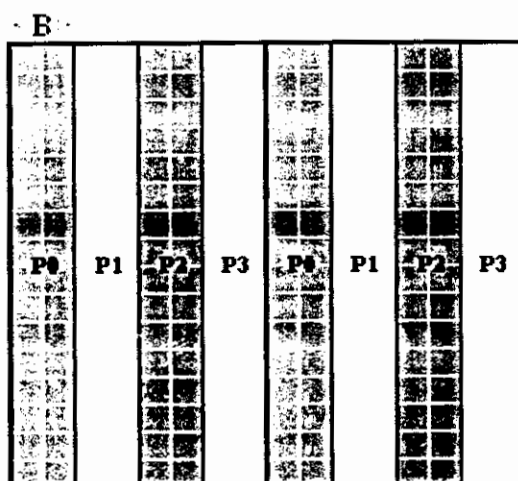
วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก [16] เป็นวิธีการจัดรูปแบบข้อมูลที่มีลักษณะการกระจายข้อมูลคล้ายกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ซึ่งมีรูปแบบการกระจายข้อมูลเป็นคอลัมน์ โดยจะแบ่งข้อมูลออกเป็นคอลัมน์ที่มีขนาด 1 คอลัมน์ ในการกระจายข้อมูลนั้นจะแบ่งข้อมูลคอลัมน์แรกให้กับหน่วยประมวลผลตัวที่ 1 และแบ่งข้อมูลคอลัมน์ถัดไปให้กับหน่วยประมวลผลตัวถัด ๆ วนกว่าจะครบตามจำนวนของหน่วยประมวลผลที่มีในระบบคลัสเตอร์ หากข้อมูลยังไม่หมดก็จะเริ่มต้นแบ่งข้อมูลให้กับหน่วยประมวลผลตัวแรกอีกครั้งและแบ่งข้อมูลคอลัมน์ถัดไปให้กับหน่วยประมวลผลตัวถัดไป โดยจะทำตามวิธีที่กล่าวมานี้จนกว่าข้อมูลจะหมด ดังแสดงในภาพที่ 2.13



ภาพที่ 2.13 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก

2.1.4.3 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก (Column Block Cyclic)

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก [16] เป็นวิธีการจัดรูปแบบข้อมูลที่ใช้ในการกระจายข้อมูลให้กับแต่ละหน่วยประมวลผลบนระบบคลัสเตอร์ ซึ่งมีรูปแบบการกระจายข้อมูลเป็นคอลัมน์ โดยจะแบ่งข้อมูลออกเป็นบล็อกของคอลัมน์ที่มีขนาดเท่ากันตามที่กำหนดขึ้น ในการแบ่งข้อมูลนั้นจะแบ่งข้อมูลบล็อกแรกให้กับหน่วยประมวลผลตัวที่ 1 และแบ่งข้อมูลบล็อกถัดไปให้กับหน่วยประมวลผลตัวถัด ๆ เมื่อทำการแบ่งข้อมูลครบรอบตามจำนวนของหน่วยประมวลผลทั้งหมดที่มีบนระบบคลัสเตอร์ หากข้อมูลยังไม่หมดจะเริ่มต้นแบ่งข้อมูลให้กับหน่วยประมวลผลตัวแรกอีกครั้งและแบ่งข้อมูลบล็อกถัดไปให้กับหน่วยประมวลผลตัวถัดไป ซึ่งจะทำตามวิธีที่กล่าวมานี้จนกว่าข้อมูลจะหมด ดังแสดงในภาพที่ 2.13



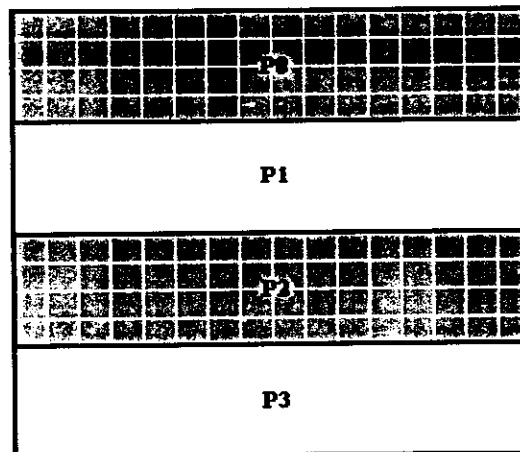
ภาพที่ 2.14 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก

2.1.4.4 วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อก (Row Block)

วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อก [16] เป็นวิธีการจัดรูปแบบข้อมูลที่ใช้ในการกระจายข้อมูล ซึ่งมีลักษณะการแบ่งข้อมูลเหมือนกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกเพียงแต่เปลี่ยนจากการแบ่งเป็นคอลัมน์มาเป็นแถว โดยจะแบ่งข้อมูลออกเป็นบล็อกของแถวที่มีขนาดเท่ากัน ในแต่ละบล็อกมีขนาดเท่ากับ n/p เมื่อ n หมายถึง ขนาดของเมตริกซ์ และ p หมายถึง จำนวนหน่วยประมวลผลทั้งหมดที่มีในระบบคลัสเตอร์ ในการกระจายข้อมูลให้กับแต่ละหน่วยประมวลผลนั้น ข้อมูลบล็อกแรกจะถูกแบ่งให้กับหน่วยประมวลผลตัวแรก ข้อมูลบล็อกถัดไปจะถูกแบ่งให้กับหน่วยประมวลผลตัวถัดไป โดยจะทำอย่างนี้ไปเรื่อย ๆ จนกว่าหน่วยประมวลผลทุกตัวจะได้รับข้อมูลจนครบ ดังแสดงในภาพที่ 2.15

ชท/วพ
๑A
๙287

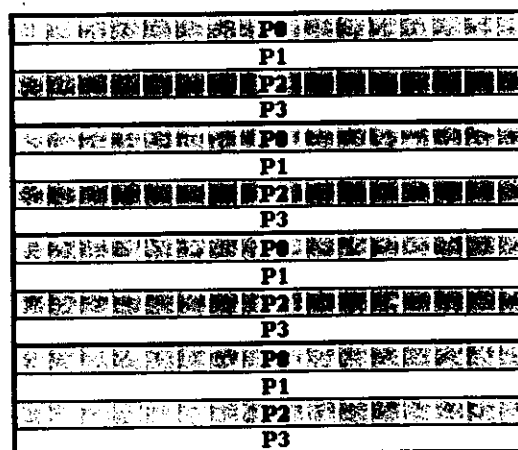
๖๓๖๑๗
ข้อมูลท้องถิ่น



ภาพที่ 2.15 วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อก

2.1.4.5 วิธีการจัดรูปแบบข้อมูลแบบแถวไซคลิก (Row Cyclic)

วิธีการจัดรูปแบบข้อมูลแบบแถวไซคลิก [17] เป็นวิธีการจัดรูปแบบข้อมูลที่ใช้ในการกระจายข้อมูล ซึ่งมีลักษณะการแบ่งข้อมูลเหมือนกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก เพียงแต่เปลี่ยนจากการแบ่งเป็นคอลัมน์มาเป็นแถว ซึ่งมีรูปแบบการกระจายข้อมูลเป็นแถว โดยจะแบ่งข้อมูลออกเป็นแถวที่มีขนาด 1 แถว ในการกระจายข้อมูลนั้นจะแบ่งข้อมูลแถวแรกให้กับหน่วยประมวลผลตัวที่ 1 และแบ่งข้อมูลแถวถัดไปให้กับหน่วยประมวลผลตัวถัด ๆ จนกว่าจะครบตามจำนวนของหน่วยประมวลผลที่มีในระบบคลัสเตอร์ หากข้อมูลยังไม่หมดก็จะเริ่มต้นแบ่งข้อมูลให้กับหน่วยประมวลผลตัวแรกอีกครั้งและแบ่งข้อมูลแถวถัดไปให้กับหน่วยประมวลผลตัวถัดไป โดยจะทำตามวิธีที่กล่าวมานี้จนกว่าข้อมูลจะหมด ดังแสดงในภาพที่ 2.16

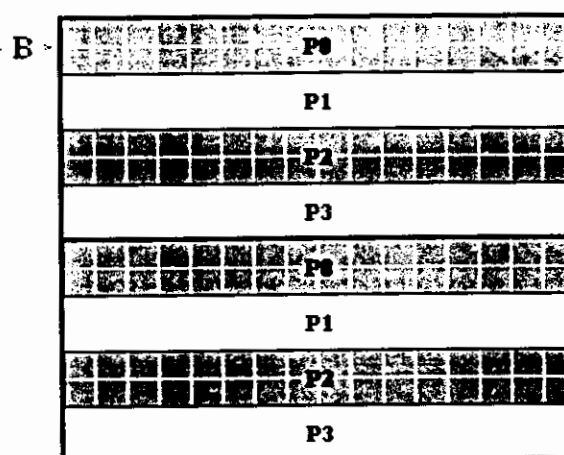


ภาพที่ 2.16 วิธีการจัดรูปแบบข้อมูลแบบแถวไซคลิก

ใช้เฉพาะใน
ศูนย์ข้อมูลท้องถิ่นเท่านั้น

2.1.4.6 วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อกไซคลิก (Row Block Cyclic)

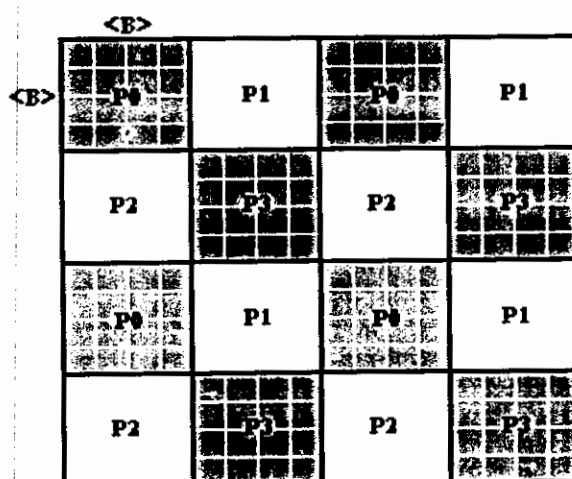
วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อกไซคลิก เป็นวิธีการจัดรูปแบบข้อมูลที่ใช้ในการกระจายข้อมูลให้กับแต่ละโหนดบนระบบคลัสเตอร์ ซึ่งมีรูปแบบการกระจายข้อมูลเป็นแถว โดยจะแบ่งข้อมูลออกเป็นบล็อกของแถวที่มีขนาดเท่ากันตามที่กำหนดขึ้น ในการแบ่งข้อมูลนั้นจะแบ่งข้อมูลบล็อกแรกให้กับหน่วยประมวลผลตัวที่ 1 และแบ่งข้อมูลบล็อกถัดไปให้กับหน่วยประมวลผลตัวถัด ๆ เมื่อทำการแบ่งข้อมูลครบรอบตามจำนวนของหน่วยประมวลผลทั้งหมดที่มีบนระบบคลัสเตอร์ หากข้อมูลยังไม่หมดจะเริ่มต้นแบ่งข้อมูลให้กับหน่วยประมวลผลตัวแรกอีกครั้งและแบ่งข้อมูลบล็อกถัดไปให้กับหน่วยประมวลผลตัวถัดไป ซึ่งจะทำตามวิธีที่กล่าวมานี้จนกว่าข้อมูลจะหมด ดังแสดงในภาพที่ 2.17



ภาพที่ 2.17 วิธีการจัดรูปแบบข้อมูลแบบแถวบล็อกไซคลิก

2.1.4.7 วิธีการจัดรูปแบบข้อมูลแบบแถวและคอลัมน์บล็อกไซคลิก (Row and Column Block Cyclic)

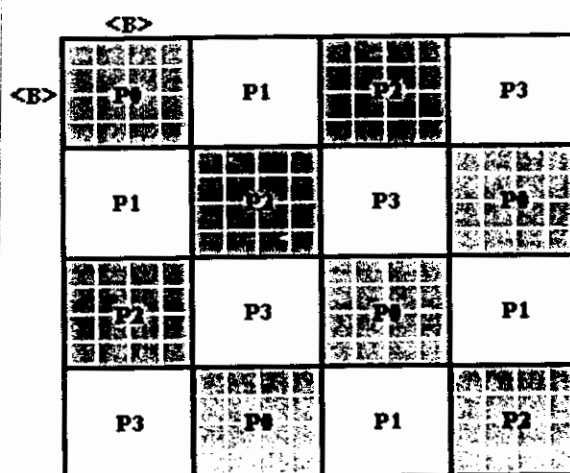
วิธีการจัดรูปแบบข้อมูลแบบแถวและคอลัมน์บล็อกไซคลิก [16, 18] ซึ่งเป็นวิธีการจัดรูปแบบข้อมูลที่ใช้ในการกระจายข้อมูลให้กับแต่ละหน่วยประมวลผลบนระบบคลัสเตอร์ที่มีลักษณะแบบ 2 มิติ (Two Dimensions) โดยจะแบ่งข้อมูลออกเป็นบล็อกของแถวและบล็อกของคอลัมน์ โดยขนาดของแถวและคอลัมน์ในแต่ละบล็อกนั้นขึ้นอยู่กับขนาดที่กำหนดขึ้น ในแต่ละบล็อกจะต้องอยู่ในรูปสี่เหลี่ยมด้านเท่า คือมีจำนวนของแถวและคอลัมน์เท่า ๆ กันในแต่ละบล็อก ดังแสดงในภาพที่ 2.18



ภาพที่ 2.18 วิธีการจัดรูปแบบข้อมูลแบบแถวและคอลัมน์บล็อกไซคลิก

2.1.4.8 วิธีการจัดรูปแบบข้อมูลแบบสควิเอปบล็อก (Skewed Block)

วิธีการจัดรูปแบบข้อมูลแบบสควิเอปบล็อก [16] เป็นวิธีการจัดรูปแบบข้อมูลที่ใช้ในการกระจายข้อมูลที่มีลักษณะแบบ 2 มิติ (Two Dimensions) ในกรณีที่ระบบคลัสเตอร์ ประกอบด้วย โหนดทั้งหมด 4 โหนด โดยแต่ละโหนดมีหน่วยประมวลผล 1 หน่วยประมวลผล จะทำการแบ่งข้อมูลเมตริกซ์ออกเป็นบล็อกของแถวและบล็อกของคอลัมน์ โดยในแถวแรกจะแบ่งคอลัมน์บล็อกแรกให้กับหน่วยประมวลผลตัวที่ 1 ในแถวที่ 2 จะแบ่งคอลัมน์บล็อกแรกให้กับหน่วยประมวลผลตัวที่ 2 ในแถวที่ 3 จะแบ่งคอลัมน์บล็อกแรกให้กับหน่วยประมวลผลตัวที่ 3 และในคอลัมน์สุดท้ายก็จะแบ่งข้อมูลคอลัมน์บล็อกแรกให้กับหน่วยประมวลผลตัวสุดท้าย ดังแสดงในภาพที่ 2.19



ภาพที่ 2.19 วิธีการจัดรูปแบบข้อมูลแบบสควิเอปบล็อก

2.1.5 การประมวลผลแบบขนาน (Parallel Computing)

การประมวลผลแบบขนาน [2] คือ การที่ให้คอมพิวเตอร์หลาย ๆ เครื่องช่วยกันประมวลผลการทำงานเดียวกัน โดยจะอาศัยกระบวนการและการโปรแกรมต่าง ๆ เข้ามาช่วยในการควบคุม การแจกแจงงานให้คอมพิวเตอร์แต่ละตัวอย่างมีประสิทธิภาพ และจะต้องใช้เครือข่ายที่มีประสิทธิภาพ การประมวลผลแบบขนานสามารถแบ่งออกได้ 3 ประเภทดังนี้

2.1.5.1 ระบบหลายหน่วยประมวลผลหน่วยความจำร่วม

คอมพิวเตอร์ทั่วไปประกอบด้วยหน่วยประมวลผลการทำงานตามโปรแกรมที่จัดเก็บไว้ในหน่วยความจำหลัก ซึ่งแต่ละที่ตั้งของหน่วยความจำในคอมพิวเตอร์ทั้งหมดถูกกำหนดที่ตั้ง (location) โดยตัวเลขที่เรียกว่า ที่อยู่ (address) ซึ่งเริ่มต้นที่ 0 ไปถึง 2^n เมื่อมีขนาดของที่อยู่เท่ากับ n บิต วิธีขยายโครงสร้างหน่วยประมวลผลเดี่ยว (single processor model) [2] ไปเป็นโครงสร้างหลายหน่วยประมวลผล (multiple processor) ที่ติดต่อกับหลายส่วนของหน่วยความจำ โดยแต่ละหน่วยประมวลผลสามารถเข้าถึงที่อยู่ของหน่วยความจำ โครงสร้างที่เรียกว่า หน่วยความจำที่ใช้ร่วมกัน (shared memory) การติดต่อระหว่างหน่วยประมวลผลกับหน่วยความจำจะผ่านทางรูปแบบของระบบเครือข่ายเชื่อมโยงระหว่างกัน (interconnection network) ระบบหลายหน่วยประมวลผลจะใช้ระบบที่อยู่เดียว (single address space) แต่ละที่ตั้งในระบบหน่วยความจำทั้งหมดจะมีที่อยู่เดียวและที่อยู่นี้ถูกใช้ โดยแต่ละหน่วยประมวลผลเข้าถึงที่ตั้ง เราเรียกวิธีการนี้ว่า ระบบหลายหน่วยประมวลผลหน่วยความจำร่วม (Shared Memory Multiprocessor System)

2.1.5.2 ระบบหลายหน่วยประมวลผลแบบส่งข้อความ

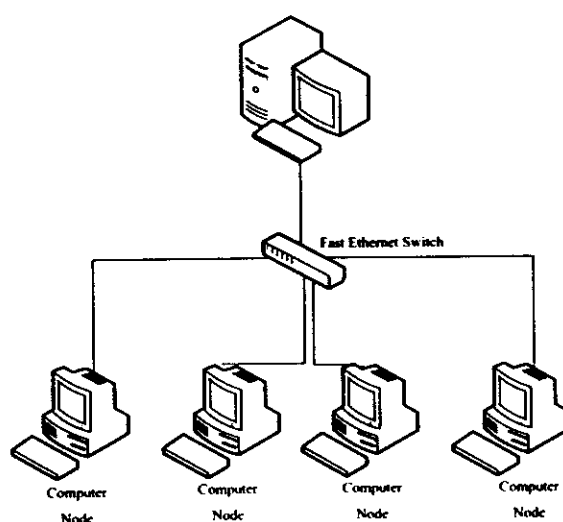
ระบบหลายหน่วยประมวลผลหน่วยความจำร่วม [2] เป็นระบบที่ถูกออกแบบมาเป็นพิเศษ สามารถสร้างโดยการเชื่อมต่อคอมพิวเตอร์ทั้งหมดผ่านระบบเครือข่ายเชื่อมโยงระหว่างกันของคอมพิวเตอร์แต่ละเครื่อง ซึ่งในแต่ละเครื่องประกอบด้วยหน่วยประมวลผล และหน่วยความจำท้องถิ่น (local memory) ที่คอมพิวเตอร์เครื่องอื่นไม่สามารถใช้ได้ ในระบบหลายหน่วยประมวลผลหน่วยความจำถูกกระจายไปยังคอมพิวเตอร์ และแต่ละเครื่องมีแอดเดรสพิเศษของมันเอง การเข้าถึงหน่วยประมวลผลสามารถเข้าถึงได้เพียงหน่วยความจำของมันเท่านั้น ระบบเครือข่ายเชื่อมโยงระหว่างกันถูกเตรียมไว้สำหรับให้หน่วยประมวลผลส่งข้อความไปยังหน่วยประมวลผลตัวอื่น ข้อความเหล่านี้จะถูกรวบรวมที่หน่วยประมวลผลของคอมพิวเตอร์เครื่องอื่น ๆ ที่ต้องการข้อมูลเพื่อใช้ในการคำนวณของมัน ระบบมัลติคอมพิวเตอร์แบบนี้โดยทั่วไปเรียกว่า ระบบหลายคอมพิวเตอร์แบบส่งข้อความ (Message Passing Multicomputer) [2] หรือเรียกอย่างง่ายว่า มัลติคอมพิวเตอร์ (Multicomputer)

2.1.5.3 ระบบหน่วยความจำร่วมแบบกระจาย

โครงสร้างแบบหน่วยความจำร่วม (shared memory paradigm) [2] เป็นที่ต้องการในมุมมองของการเขียนโปรแกรม นักวิจัยหลาย ๆ คน ได้ติดตามแนวความคิดของระบบหน่วยความจำร่วมแบบกระจาย (distributed shared memory system) แต่ละหน่วยประมวลผลจะเข้าถึงหน่วยความจำทั้งหมด โดยใช้หน่วยความจำแบบแอดเดรสเพียงเดียว (single memory address space) สำหรับหน่วยประมวลผลที่เข้าถึงที่ตั้งของหน่วยความจำที่ไม่ได้อยู่ที่ตัวมันเอง การส่งข้อความ (message passing) ต้องเกิดขึ้น โดยส่งข้อมูลจากหน่วยประมวลผลไปยังตำแหน่งที่ต้องการหรือจากตำแหน่งที่ต้องการกลับมายังหน่วยประมวลผลในบางวิธีที่อัตโนมัติจะซ่อนความจริงที่ว่าหน่วยความจำกระจายกันอยู่ แนวความคิดนี้ถูกเรียกว่าหน่วยความจำร่วมเสมือน (shared virtual memory) ที่จะให้ภาพเสมือนของหน่วยความจำร่วม แม้ว่าจริง ๆ แล้วกระจายกันซึ่งก็มีข้อเสียในเรื่องของการเข้าถึงระยะไกล (remote access) จะทำให้เสียเวลา (delay) มากกว่าการเข้าถึงข้อมูลในเครื่องตนเอง ปัญหาดังกล่าวนี้ขียนำเอาตารางค้นหาใช้ในการหาตำแหน่งที่ตั้งที่แท้จริงในหน่วยความจำแบบกระจาย

2.1.6 ระบบคลัสเตอร์ (Cluster Computer)

ระบบคลัสเตอร์ [1] เป็นการนำเอาเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal Computer) มาต่อพ่วงกันหลาย ๆ ตัว บนเครือข่ายคอมพิวเตอร์ที่มีความเร็วสูง เพื่อใช้ประโยชน์จากหน่วยประมวลผลของคอมพิวเตอร์แต่ละเครื่อง ด้วยการแบ่งงานให้หน่วยประมวลผลแต่ละตัวทำงาน ทำให้สามารถคำนวณงานขนาดใหญ่ได้เร็วขึ้นโดยไม่จำเป็นต้องใช้ซูเปอร์คอมพิวเตอร์



ภาพที่ 2.20 ระบบคลัสเตอร์ที่มีขนาด 4 หน่วยประมวลผลเชื่อมโยงผ่านเครือข่ายความเร็วสูง

สถาปัตยกรรมของคลัสเตอร์นั้น จะประกอบด้วยเครื่องพีซีคอมพิวเตอร์หลาย ๆ ตัว ซึ่งเชื่อมต่อเข้าด้วยกันโดยใช้เครือข่ายความเร็วสูง โดยเครื่องคอมพิวเตอร์แต่ละเครื่องที่นำมาต่อเชื่อมในระบบคลัสเตอร์นี้จะเรียกว่า โหนด (Node) โดยจะมี Frontend Node เป็นโหนดหลักสำหรับการควบคุมระบบ และ Compute Node เป็นโหนดสำหรับการประมวลผลเป็นหลัก ในแต่ละโหนดจะใช้ซอฟต์แวร์แบบ Full Multitask และจะสร้างภาพเสมือนว่าเป็นเครื่องเดียวกันหมด ไม่ว่าผู้จะใช้จะเข้าใช้งานเครื่องใดก็ตาม โดยการใช้วิธีการต่าง ๆ ภายในระบบจะมีซอฟต์แวร์สำหรับใช้ในการประมวลผลแบบขนาน เช่น PVM (Parallel Virtual Machine) [19] หรือ MPI (Message Passing Interface) [20]

แนวความคิดทางสถาปัตยกรรมของคลัสเตอร์ [1] คลัสเตอร์โหนด (Cluster Node) แต่ละโหนดเป็นคอมพิวเตอร์ที่สมบูรณ์หมายความว่าแต่ละโหนดมี หน่วยประมวลผล (processor) แคช (cache) หน่วยความจำ (memory) ดิสก์ (disk) และอแดปเตอร์ในการรับข้อมูลและแสดงผล (I/O adapter) มากไปกว่านั้นคือมีระบบปฏิบัติการมาตรฐาน (standard operating system) ที่สมบูรณ์ (complete) อยู่ในแต่ละเครื่อง โหนดสามารถมีได้มากกว่าหนึ่งหน่วยประมวลผลแต่มีภาพเพียงหนึ่งสำเนาของระบบปฏิบัติการ (one copy of the OS image)

2.1.7 MPI (Message Passing Interface)

MPI (Message Passing Interface) [21] เป็น Library มาตรฐานที่ใช้ในการเขียนโปรแกรมแบบขนานที่นิยมใช้กันอย่างกว้างขวาง โดย MPI เป็น Library ที่มีลักษณะเหมาะในการเขียนโปรแกรมเพื่อส่งข้อความแบบขนานที่ทำงานบนเครื่องกระจายหน่วยความจำ โปรแกรมการส่งข้อความจะสร้างงานขึ้นมาหลาย ๆ งาน แต่ละงานจะมีค่าเอกลักษณ์ด้วยชื่อเฉพาะและมีการห่อหุ้มข้อมูลด้วยงานต่าง ๆ จะมีการส่งและการรับข้อความ ด้วยการใช้อ้างอิงโปรแกรมที่ใช้เขียนการส่งผ่านข้อมูลนั้นจะเขียนด้วยภาษาพื้นฐาน เช่น ภาษา C, C++ และ Fortran และเรียกใช้ Library ของการส่งและการรับข้อความของ MPI สำหรับเวอร์ชัน 1 ของ MPI นั้นมีแจกจ่ายในปี 1994 [20] นับตั้งแต่นั้นเป็นต้นมา MPI ก็ได้กลายมาเป็น Library มาตรฐานที่ใช้ในการเขียนโปรแกรมบนระบบขนานและมีการนำไปใช้งานกันอย่างกว้างขวางไม่ว่าจะเป็นมหาวิทยาลัย ศูนย์วิจัย ห้องปฏิบัติการระบบชาติ หรือแม้แต่ผู้จำหน่ายคอมพิวเตอร์ประสิทธิภาพสูงที่ทำงานแบบขนานส่วนใหญ่มักจะรวมเอา MPI เข้าเป็นส่วนหนึ่งของระบบปฏิบัติการของคอมพิวเตอร์แบบขนานนั้นด้วย จุดมุ่งหมายหลักของการสร้าง MPI ขึ้นมานั้นเพื่อให้เป็นภาษาในการติดต่อสื่อสารที่เป็นสากลบนระบบขนาน ซึ่ง MPI ที่ได้มีการเผยแพร่มีทั้งจะต้องเสียค่าลิขสิทธิ์ในการใช้งานและแบบที่ให้ใช้งานได้ฟรี สำหรับ MPI ที่ให้ใช้งานได้ฟรีได้แก่ LAM [21] และ MPICH [22] เป็นต้น

2.2 งานวิจัยที่เกี่ยวข้อง

จากที่ได้กล่าวไปแล้วว่าวิธีการกำจัดของเกาส์เป็นวิธีที่ใช้ในการแก้ปัญหาระบบสมการเชิงเส้นที่มีขั้นตอนการทำงานเป็นอิสระและมีการทำงานแบบไม่ซับซ้อนมาก จึงนิยมนำมาใช้ในการวิจัยต่าง ๆ ซึ่งจุดประสงค์ของการวิจัยก็มุ่งเน้นในหลายรูปแบบ เช่น การนำเอาวิธีการกำจัดของเกาส์ไปทำงานกับวิธีการแก้ปัญหาคณิตศาสตร์วิธีอื่น ๆ (LU Decomposition, Doolittle, Cholesky factorization, Crout Decomposition, Gauss-Jacobi, Gauss-Seidel, Successive over Relaxation (SOR), QR factorization), การออกแบบวิธีการจัดรูปแบบข้อมูลเพื่อนำไปเปรียบเทียบกับวิธีการจัดรูปแบบข้อมูลอื่น ๆ เพื่อให้ทราบถึงประสิทธิภาพในการทำงานของแต่ละวิธีการ และการพัฒนารูปแบบของขั้นตอนวิธี (Algorithm) ของวิธีการกำจัดของเกาส์เพื่อให้ทำงานได้อย่างมีประสิทธิภาพสูงสุด สำหรับงานวิจัยที่เรานำมาศึกษาจะศึกษานั้นจะเกี่ยวกับเรื่องของวิธีการจัดรูปแบบข้อมูลโดยใช้วิธีการแก้ปัญหาระบบสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์ ซึ่งมีนักวิจัยที่ให้ความสนใจในเรื่องนี้พอสมควร ดังนี้

James Demmel [16] ได้นำเสนอวิธีการจัดรูปแบบข้อมูลเมื่อนำมาใช้แก้ปัญหาระบบสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์ 5 รูปแบบดังนี้ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไจคลิก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไจคลิก, วิธีการจัดรูปแบบข้อมูลแบบแถวและคอลัมน์บล็อกไจคลิก และวิธีการจัดรูปแบบข้อมูลแบบสควเอบล็อก โดย James พบว่า วิธีการจัดรูปแบบข้อมูลแบบแถวและคอลัมน์บล็อกไจคลิกเป็นวิธีการจัดรูปแบบที่มีประสิทธิภาพในการทำงานสูงสุดและยังเป็นวิธีที่ใช้งานทรัพยากรได้อย่างคุ้มค่ามาก แต่มีปัญหาในเรื่องของการสลับแถวข้อมูลเมื่อนำมาใช้งานวิธีการกำจัดของเกาส์ เพราะทำให้เกิดการสื่อสารเกิดขึ้นในปริมาณที่มาก อันดับที่ 2 คือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไจคลิก อันดับที่ 3 คือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไจคลิก ทั้งสองวิธีนี้พัฒนารูปแบบการกระจายข้อมูลมาจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ซึ่งสามารถทำงานได้เร็วกว่าวิธีการจัดรูปแบบแบบคอลัมน์บล็อกเล็กน้อย โดยได้แก้ปัญหาในเรื่องของปริมาณการคำนวณและการใช้งานทรัพยากรให้คุ้มค่ามากขึ้น อันดับที่ 4 คือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ซึ่งถือว่าเป็นวิธีพื้นฐานที่ทำงานแบบคอลัมน์ แต่มีปัญหาในเรื่องของปริมาณการคำนวณและการใช้งานทรัพยากรให้คุ้มค่าเมื่อทำงานในบล็อกแรกเสร็จ หน่วยประมวลผลที่ 1 จะว่างลง และหากทำงานในบล็อกถัด ๆ ไปเสร็จ หน่วยประมวลผลตัวถัด ๆ ไปก็จะว่างลงเรื่อย ๆ ซึ่งถือว่าเป็นการใช้ทรัพยากรที่ไม่คุ้มค่าเท่าที่ควร ส่วนวิธีสุดท้ายที่ได้นำเสนอ คือ วิธีการจัดรูปแบบข้อมูลแบบสควเอบล็อก เป็นวิธีที่ไม่เหมาะในการนำมาใช้งานกับวิธีการกำจัดของเกาส์เพราะการทำงานไม่มีประสิทธิภาพเท่าที่ควร ซึ่งวิธีการนี้ถูกนำเอาไปใช้แก้ปัญหาคณิตศาสตร์ด้วยวิธีการแบบอื่น

Michael Wasilewski [7] ได้นำเอาวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไขว้คลิกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกมาเปรียบเทียบเพื่อหาประสิทธิภาพของการทำงานโดยใช้วิธีการกำจัดของเกาส์บนระบบคลัสเตอร์ โดย Michael ได้สรุปเอาไว้ว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกมีประสิทธิภาพสูงสุด วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไขว้คลิกเป็นอันดับสอง และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกเป็นอันดับสุดท้าย โดยประสิทธิภาพที่ได้แตกต่างกันไม่มากนัก อย่างที่ได้นำเสนอไปแล้วว่า ถึงแม้จะสามารถลดปริมาณการคำนวณลงและทำให้ใช้งานทรัพยากรได้อย่างคุ้มค่าขึ้นแต่ก็มีปัญหาในเรื่องของปริมาณการคำนวณที่มีเพิ่มมากขึ้นด้วย หากมีวิธีการจัดรูปแบบข้อมูลที่สามารถลดปริมาณการคำนวณลงให้น้อยกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกโดยไม่ทำให้เกิดการสื่อสารเพิ่มขึ้นมากนัก น่าจะทำให้การทำงานได้อย่างมีประสิทธิภาพมากขึ้น

บทที่ 3

การวิเคราะห์และออกแบบ

ในบทนี้ เนื้อหาประกอบด้วย 3 ส่วนคือ ส่วนที่ 1 การประมวลผลแบบลำดับเป็นการศึกษารูปแบบการประมวลผลแบบลำดับของวิธีการกำจัดของเกาส์ ส่วนที่ 2 วิธีการจัดรูปแบบข้อมูลแบบขนานบนเครื่องกระจายหน่วยความจำ ศึกษาวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก (Column Block) และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก (Column Block Cyclic) จากนั้นทำการออกแบบวิธีการจัดรูปแบบข้อมูลใหม่โดยศึกษาด้านแบบจากวิธีการจัดรูปแบบข้อมูลทั้งสองวิธี โดยวิธีการใหม่นี้เรียกว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก (Column Block and Column Cyclic) โดยในแต่ละวิธีได้แยกออกเป็น 2 แบบคือ แบบสลับแถว (Pivoting) และแบบไม่สลับแถว (Without Pivoting) ส่วนที่ 3 การวิเคราะห์เวลาในการทำงาน เป็นส่วนที่ศึกษาปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลในแต่ละแบบเพื่อให้เห็นถึงประสิทธิภาพในการทำงาน

3.1 การประมวลผลแบบลำดับ

การประมวลผลแบบลำดับ (Sequential) เป็นการทำงานโดยกระบวนการแก้ปัญหาสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์ทั้งหมดเกิดขึ้นที่เครื่องคอมพิวเตอร์เพียงเครื่องเดียว ข้อมูลที่นำมาใช้ในการประมวลผลนั้นถูกแบ่งให้กับคอมพิวเตอร์เพียงเครื่องเดียวเท่านั้น ตัวอย่างเช่น ข้อมูลเมตริกซ์ A ขนาด 8×8 เมตริกซ์ B ขนาด 8×1 เมื่อต้องการนำมาประมวลผล ได้กระจายข้อมูลให้กับเครื่องคอมพิวเตอร์หนึ่งเครื่อง ดังแสดงในภาพที่ 3.1

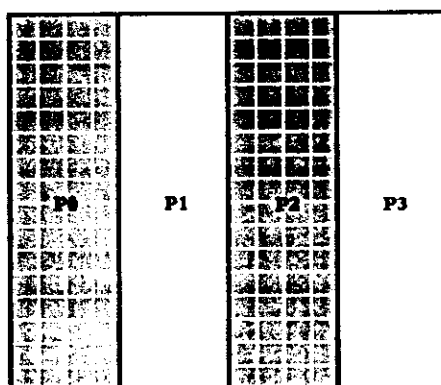
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

ภาพที่ 3.1 การกระจายข้อมูลแบบลำดับ เมตริกซ์ A ขนาด 8×8 เมตริกซ์ B ขนาด 8×1

3.2 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกเป็นวิธีการพื้นฐานของการกระจายข้อมูลในการทำงานแบบขนานที่นิยมใช้กันในงานวิจัยต่าง ๆ อย่างที่ได้กล่าวไปแล้วในบทที่ 2 ในเรื่องการทำงานของวิธีการกำจัดของเกาส์ที่ต้องมีการสลับแถวเพื่อลดข้อผิดพลาดในการทำงาน วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกจึงเหมาะสมที่จะนำมาใช้ หากกระบวนการขั้นตอนใดขั้นตอนหนึ่งมีการสลับแถวของข้อมูลไม่เกิดการสื่อสารที่มากจนเกินไป เพราะการตรวจสอบหาค่าสมบูรณ์สูงสุดมีในการทำงานแบบคอลัมน์ ซึ่งในแต่ละคอลัมน์อยู่ภายในหน่วยประมวลผลเพียงหน่วยเดียว ถ้าหากจำเป็นต้องสลับแถวข้อมูลก็เป็นการสลับข้อมูลภายในตัวเองจึงไม่เกิดการสื่อสาร การสื่อสารที่เกิดขึ้นในกรณีที่มีการสลับแถวข้อมูลก็เป็นเพียงการส่งข้อความไปแจ้งหน่วยประมวลผลที่อยู่ถัดจากหน่วยประมวลผลหลักที่มีการสลับแถวเท่านั้น

วิธีการจัดรูปแบบข้อมูลแบบนี้ ได้ทำการกระจายข้อมูลให้แต่ละหน่วยประมวลผลเป็นแบบบล็อกของคอลัมน์ โดยแต่ละหน่วยประมวลผลที่อยู่ในระบบคลัสเตอร์จะได้รับข้อมูลที่เป็นบล็อกของคอลัมน์ที่มีขนาดเท่ากัน ซึ่งขนาดของบล็อกนั้นได้มาจากการหารจำนวนของเมตริกซ์ A ด้วยจำนวนของหน่วยประมวลผลที่มีทั้งหมดในระบบคลัสเตอร์ รูปแบบการกระจายข้อมูลของวิธีการนี้แสดงในภาพที่ 3.2



ภาพที่ 3.2 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกระบบคลัสเตอร์ 4 หน่วยประมวลผล

หลังจากทำการแบ่งข้อมูลออกในสัดส่วนที่เท่ากัน จากนั้นกระจายข้อมูลไปให้กับหน่วยประมวลผลต่าง ๆ ที่มีอยู่ในระบบคลัสเตอร์ ตัวอย่างเช่น เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล เมื่อกระจายข้อมูลให้แต่ละหน่วยประมวลผลที่ได้เป็นดังภาพที่ 3.3

0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3
0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	3

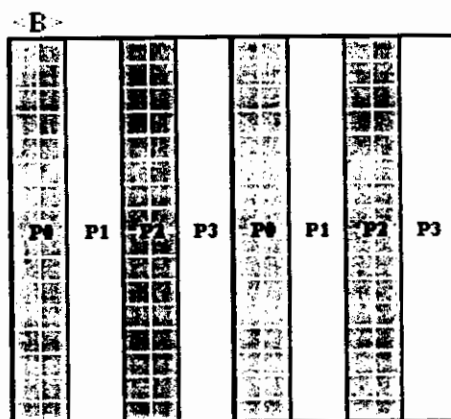
ภาพที่ 3.3 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1
บนระบบคลัสเตอร์ที่มี 4 หน่วยประมวลผล

3.3 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก เป็นวิธีการจัดรูปแบบข้อมูลที่ออกแบบมาเพื่อทำงานแบบขนาน ซึ่งพัฒนามาจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกนั้นมีปัญหาในเรื่องของปริมาณของการคำนวณและการใช้ทรัพยากร ไม่คุ้มค่า ถ้าหากทำงานเสร็จในแต่ละบล็อกหน่วยประมวลผลที่รับผิดชอบในบล็อกนั้น ๆ ก็จะมีว่างลง ดังนั้นจึงได้ออกแบบวิธีการจัดรูปแบบข้อมูลแบบนี้ขึ้นมาเพื่อลดปริมาณของการคำนวณลง และสามารถใช้งานทรัพยากรที่มีอยู่ได้อย่างคุ้มค่ามากขึ้น ทำให้สามารถทำงานได้เร็วขึ้น แต่วิธีการนี้ [17] ก็มีข้อเสียตรงที่มีปริมาณของการสื่อสารเพิ่มมากขึ้นด้วย

สำหรับรูปแบบการกระจายข้อมูลของวิธีการนี้ แบ่งออกข้อมูลออกเป็นบล็อกที่มีขนาดเท่า ๆ กันตามขนาดที่กำหนดขึ้นจากนั้นก็แบ่งให้แต่ละหน่วยประมวลผลที่มีอยู่ในระบบคลัสเตอร์ โดยแบ่งข้อมูลบล็อกแรกให้กับหน่วยประมวลผลตัวแรก แบ่งข้อมูลบล็อกถัดไปให้กับหน่วยประมวลผลตัวถัดไปจนกว่าหน่วยประมวลผลทุกตัวได้รับข้อมูลจนครบ หากข้อมูลยังไม่หมดก็เริ่มแบ่งข้อมูลให้กับหน่วยประมวลผลตัวแรกใหม่อีกครั้งจนกว่าจะครบทุกตัว จะทำอย่างนี้ไปจนกว่าข้อมูลจะหมด ขนาดของแต่ละบล็อกที่แบ่งนั้น จะส่งผลต่อประสิทธิภาพในการทำงาน

การแบ่งข้อมูลออกเป็นบล็อกที่มีขนาดที่แตกต่างกันนั้นเมื่อนำไปใช้งาน จะมีปริมาณการคำนวณและการสื่อสารที่แตกต่างกัน รูปแบบการกระจายข้อมูลของวิธีการนี้ดังแสดงในภาพที่ 3.4



ภาพที่ 3.4 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ตัวอย่างเช่น เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล สามารถกระจายข้อมูลได้ดังแสดงในภาพที่ 3.5

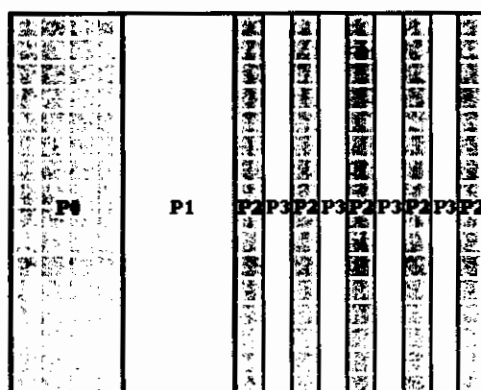
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3
0	0	1	1	2	2	3	3	0	0	1	1	2	2	3	3	3

ภาพที่ 3.5 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์และคอลัมน์ไซคลิก เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

3.4 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก เป็นวิธีการจัดรูปแบบข้อมูลที่ออกแบบมาเพื่อทำงานแบบขนาน ซึ่งออกแบบขึ้นใหม่ในการทำวิจัยในครั้งนี้ การออกแบบได้อาศัยวิธีการจัดรูปแบบข้อมูลต้นแบบ 2 รูปแบบวิธี คือวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกที่มีรูปแบบการทำงานที่เหมาะสมกับวิธีการกำจัดของเกาส์อยู่แล้วแต่มีปัญหาในเรื่องของปริมาณของการคำนวณและความคุ้มค่าในการใช้งานหน่วยประมวลผล และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกซึ่งออกแบบมาเพื่อแก้ปัญหของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกแต่ก็มีปัญหาในเรื่องของปริมาณการสื่อสารที่เพิ่มมากขึ้น เพราะฉะนั้นจึงได้ออกแบบวิธีการจัดรูปแบบข้อมูลใหม่ที่สามารถลดปริมาณของการคำนวณลงได้พอสมควรและไม่เพิ่มปริมาณของการสื่อสารมากเกินไป ซึ่งน่าจะทำให้สามารถทำงานได้เร็วขึ้น

สำหรับรูปแบบการกระจายข้อมูลของวิธีการนี้ ต้องแบ่งข้อมูลออกเป็น 2 คอลัมน์และแบ่งหน่วยประมวลผลที่มีอยู่ในระบบคลัสเตอร์ออกเป็น 2 กลุ่มเท่า ๆ กัน ในคอลัมน์แรกก็จะแบ่งข้อมูลย่อยออกเป็นบล็อกของคอลัมน์ ตามจำนวนของหน่วยประมวลผลกลุ่มแรก โดยบล็อกของคอลัมน์แรกจะแบ่งให้กับหน่วยประมวลผลตัวที่ 1 บล็อกถัดไปจะแบ่งให้หน่วยประมวลผลตัวถัดไปที่อยู่ในกลุ่มของหน่วยประมวลผลชุดแรกจนกว่าจะครบทุกตัว ส่วนในคอลัมน์ที่ 2 จะแบ่งข้อมูลออกเป็นคอลัมน์ที่มีขนาด 1 คอลัมน์แล้วทำการแบ่งข้อมูลให้กับหน่วยประมวลผลชุดที่ 2 โดยข้อมูลคอลัมน์แรกจะแบ่งให้กับหน่วยประมวลผลตัวแรกของชุดที่ 2 คอลัมน์ถัดไป จะแบ่งให้หน่วยประมวลผลตัวถัดไปที่อยู่ในชุดที่ 2 จะทำการแบ่งจนกว่าหน่วยประมวลผลที่อยู่ในชุดที่ 2 ได้รับข้อมูลครบทุกตัว หากข้อมูลยังไม่หมดก็จะทำการแบ่งข้อมูลให้กับหน่วยประมวลผลตัวแรกและตัวถัด ๆ ไป จนกว่าข้อมูลจะหมด ดังแสดงในภาพที่ 3.6



ภาพที่ 3.6 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ตัวอย่างเช่น เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล เมื่อทำการกระจายข้อมูลมีลักษณะดังแสดงในภาพที่ 3.7

0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2
0	0	0	0	1	1	1	1	2	3	2	3	2	3	2	3	2

ภาพที่ 3.7 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์และคอลัมน์ไซคลิก เมตริกซ์ A ขนาด 16×16 เมตริกซ์ B ขนาด 16×1 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

3.5 การวิเคราะห์เวลาในการทำงาน

เป้าหมายหลักของการนำเอาวิธีการแก้ปัญหาทางคณิตศาสตร์มาทำงานแบบขนานบนระบบคลัสเตอร์ก็คือ ต้องการประสิทธิภาพในการทำงานที่มากขึ้น สำหรับประสิทธิภาพในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบต่าง ๆ นั้น วัดจากเวลาที่ใช้ในการทำงาน ซึ่งสิ่งที่ส่งผลให้เกิดเวลาในการทำงานที่แตกต่างกันของแต่ละรูปแบบนั้น พิจารณาจากตัวชี้วัด 2 ตัว โดยตัวชี้วัดตัวแรกคือ การสื่อสารเป็นการติดต่อสื่อสารกันระหว่างหน่วยประมวลผลต่าง ๆ ที่มีอยู่ในระบบคลัสเตอร์เพื่อทำการแลกเปลี่ยนข้อมูลต่าง ๆ ระหว่างกัน โดยข้อมูลที่ใช้ในการสื่อสารจะอยู่ในรูปแบบของข้อความที่ส่งถึงกัน ส่วนตัวชี้วัดตัวที่สองคือ การคำนวณเป็นการคำนวณที่เกิดจากหน่วยประมวลผลแต่ละตัวทำการประมวลผลเพื่อกำจัดค่าสัมประสิทธิ์ของตัวแปรที่ไม่ทราบค่าของสมการ เพราะฉะนั้นสิ่งที่สนใจศึกษาคือ ปริมาณของการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแต่ละตัวว่ามีมากน้อยเพียงไร

3.5.1 การประมวลผลแบบลำดับ

การประมวลผลแบบลำดับเป็นการทำงานที่กระบวนการทั้งหมดเกิดขึ้นภายในคอมพิวเตอร์เครื่องเดียว เพราะฉะนั้นจึงไม่เกิดการสื่อสารเกิดขึ้น มีเพียงการคำนวณเท่านั้น โดยในแต่ละชุดข้อมูลมีการคำนวณในปริมาณที่แตกต่างกันขึ้นอยู่กับขนาดของเมตริกซ์ที่นำมาทดสอบ สำหรับการประมวลผลแบบลำดับนั้นได้แบ่งการทำงานออกเป็น 2 แบบคือ การประมวลผลแบบมีการสลับแถวและการประมวลผลแบบไม่มีการสลับแถว จากที่ได้กล่าวมาแล้วในบทที่ 2 ว่าการสลับแถวนั้นสามารถลดข้อผิดพลาดในการทำงานของวิธีการกำจัดของเกาส์ได้ แต่อาจทำให้ความเร็วในการทำงานลดลง เพราะต้องใช้เวลาในการตรวจสอบหาค่าสมบูรณ์สูงสุดในคอลัมน์หลักที่ทำงานอยู่และต้องใช้เวลาในการสลับแถวข้อมูล

สำหรับปริมาณของการคำนวณที่เกิดขึ้นในการทำงานของการประมวลผลแบบลำดับ ทั้งแบบมีการสลับแถวและไม่มีการสลับแถว ซึ่งทั้งสองวิธีมีปริมาณการคำนวณที่ไม่แตกต่างกัน ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 ปริมาณการคำนวณของวิธีการทำงานแบบลำดับ

ขนาดเมตริกซ์	การคำนวณ (ครั้ง)
4	46
8	364
16	2,840
32	22,320
64	176,736
128	1,406,144
256	11,217,280
512	89,608,960
1024	716,350,976
2048	5,728,717,824
4096	45,821,368,320

3.5.2 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก เป็นวิธีการจัดรูปแบบข้อมูลที่มีการทำงานแบบขนานบนระบบเครือข่ายคลัสเตอร์ ในแต่ละขั้นตอนของการกำจัดตัวแปรที่ไม่ทราบค่า จะเกิดปริมาณของการสื่อสารและการคำนวณแตกต่างกันไปขึ้นอยู่กับว่าหน่วยประมวลผลที่กำลังทำงานอยู่นั้นมีการใช้ข้อมูลร่วมกันกับหน่วยประมวลผลอื่นหรือไม่ ในการทำวิจัยครั้งนี้ได้แบ่งการทำงานออกเป็น 2 แบบเช่นเดียวกับการประมวลผลแบบลำดับ คือ การประมวลแบบมีการสลับแถว และการประมวลผลแบบไม่มีการสลับแถว การสลับแถวข้อมูลจะมีผลต่อปริมาณของการสื่อสาร ถ้าหากหน่วยประมวลผลที่มีการทำงานอยู่มีการสลับแถวต้องมีการส่งข้อความไปแจ้งให้หน่วยประมวลผลอื่น ๆ ที่อยู่ถัดไปได้ทราบว่ามีการสลับแถวของข้อมูลเกิดขึ้น ปริมาณการสื่อสารที่เกิดขึ้นในแต่ละครั้งที่มีการสลับแถวนั้นก็ขึ้นอยู่กับจำนวนหน่วยประมวลผลที่อยู่ถัดจากหน่วยประมวลผลหลักที่กำลังทำงานอยู่ว่ามีมากน้อยเพียงใด สำหรับปริมาณการสื่อสารและการคำนวณที่เกิดขึ้นในวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกสามารถดูได้จากตารางที่ 3.2

ตารางที่ 3.2 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
4	20	14	30
8	74	62	194
16	284	260	1,300
32	1,112	1,064	9,256
64	4,400	4,304	69,200
128	17,504	17,312	533,664
256	69,824	69,440	4,188,480
512	278,912	278,144	33,182,336
1024	1,114,880	1,113,344	264,152,320
2048	4,457,984	4,454,912	2,107,984,384
4096	17,828,864	17,822,720	16,842,920,960

3.5.3 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก เป็นวิธีการจัดรูปแบบข้อมูลที่ทำงานแบบขนานบนระบบเครือข่ายคลัสเตอร์ จากที่ได้กล่าวไปแล้วว่าวิธีการจัดรูปแบบข้อมูลนี้ได้อาศัยต้นแบบมาจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก แต่มีการปรับปรุงการแบ่งข้อมูลโดยมีวัตถุประสงค์เพื่อที่จะลดปริมาณการคำนวณให้น้อยลงกว่าเดิมและทำให้ใช้งานทรัพยากรให้คุ้มค่ามากขึ้น แต่ในขณะเดียวกันวิธีการนี้ก็เกิดปริมาณการสื่อสารที่เพิ่มขึ้นกว่าวิธีต้นแบบในปริมาณที่มากพอสมควร ในการทำวิจัยครั้งนี้ได้แบ่งการทำงานออกเป็น 2 แบบเช่นเดียวกับวิธีการจัดรูปแบบข้อมูลแบบอื่น ๆ คือแบบมีการสลับแถวข้อมูลและแบบไม่มีการสลับแถวข้อมูล

สำหรับปริมาณของการสื่อสารและการคำนวณที่เกิดขึ้นในวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก เมื่อแบ่งข้อมูลออกเป็นบล็อกที่มีขนาด 4, 8, 16, 32, 64, 128, 256 และ 512 สามารถดูได้จากตารางที่ 3.3 ถึงตารางที่ 3.10

ตารางที่ 3.3 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 4 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
16	284	260	1,300
32	1,460	1,388	8,444
64	6,116	5,948	48,068
128	24,644	24,284	56,588
256	98,564	97,820	3,013,740
512	393,860	392,348	23,247,084
1024	1,574,276	1,571,228	182,481,388
2048	6,294,404	6,288,284	1,445,782,508
4096	25,171,844	25,159,580	11,509,811,180

ตารางที่ 3.4 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 8 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
32	1,112	1,064	9,256
64	5,768	5,624	62,456
128	24,296	23,960	431,256
256	98,216	97,496	3,135,960
512	393,512	392,024	23,753,816
1024	1,573,928	1,570,904	184,543,576
2048	6,294,056	6,287,960	1,454,101,336
4096	25,171,496	25,159,256	11,543,226,200

ตารางที่ 3.5 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 16 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
64	4,400	4,304	69,200
128	22,928	22,640	478,704
256	96,848	96,176	3,363,120
512	392,144	390,704	24,733,616
1024	1,572,560	1,569,584	188,601,520
2048	6,292,688	6,286,640	1,470,607,024
4096	25,170,128	25,157,936	11,609,793,200

ตารางที่ 3.6 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 32 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
128	17,504	17,312	533,664
256	91,424	90,848	3,744,736
512	386,720	385,376	26,554,976
1024	1,567,136	1,564,256	196,448,096
2048	6,287,264	6,281,312	1,503,086,944
4096	25,164,704	25,152,608	11,741,871,456

ตารางที่ 3.7 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 64 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
256	69,824	69,440	4,188,480
512	365,120	363,968	29,616,064
1024	1,545,536	1,542,848	211,035,328
2048	6,265,664	6,259,904	1,565,892,288
4096	25,143,104	25,131,200	12,001,776,320

ตารางที่ 3.8 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 128 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
512	278,912	278,144	33,182,336
1024	1,459,328	1,457,024	235,556,736
2048	6,179,456	6,174,080	1,682,655,616
4096	25,056,896	25,045,376	12,504,350,080

ตารางที่ 3.9 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 256 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
1024	1,114,880	1,113,344	264,152,320
2048	5,835,008	5,830,400	1,878,957,824
4096	24,712,448	24,701,696	13,438,718,720

ตารางที่ 3.10 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
ไซคลิก บล็อกข้อมูลขนาด 512 บนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
2048	4,457,984	4,454,912	2,107,984,384
4096	23,335,424	23,326,208	15,009,660,416

3.5.4. วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก

วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก เป็นวิธีการจัดรูปแบบข้อมูลแบบใหม่ที่ทำงานแบบขนานบนระบบเครือข่ายคลัสเตอร์ จากที่ได้กล่าวไปแล้วว่าวิธีการจัดรูปแบบข้อมูลใหม่นี้ได้อาศัยค้นแบบมาจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและแบบคอลัมน์บล็อกไซคลิก แต่มีการปรับปรุงวิธีการแบ่งข้อมูล โดยมีวัตถุประสงค์เพื่อที่จะลดปริมาณการคำนวณให้น้อยลงกว่าเดิมและไม่ทำให้เกิดปริมาณการสื่อสารมากขึ้นจนเกินไป ในการทำวิจัยครั้งนี้ได้แบ่งการทำงานออกเป็น 2 แบบเช่นเดียวกับวิธีการจัดรูปแบบข้อมูลแบบอื่น ๆ คือแบบมีการสลับแถวข้อมูลและแบบไม่มีการสลับแถวข้อมูล ในแต่ละขั้นตอนของการกำจัดตัวแปรที่ไม่ทราบค่าจะเกิดการสื่อสารและการคำนวณแตกต่างกันไป โดยการทำงานแบบสลับแถวจะมีปริมาณการสื่อสารที่มากกว่าแบบที่ไม่มีการสลับแถว

สำหรับปริมาณของการสื่อสารและการคำนวณที่เกิดขึ้นในวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก สามารถดูได้จากตารางที่ 3.11

ตารางที่ 3.11 ปริมาณการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก และคอลัมน์ไซคลิกบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ขนาด เมตริกซ์	การสื่อสาร (ครั้ง)		การคำนวณ (ครั้ง)
	สลับแถว	ไม่สลับแถว	
4	20	14	28
8	76	63	182
16	293	266	1,180
32	1,147	1,092	8,536
64	4,535	4,424	64,304
128	18,031	17,808	497,760
256	71,903	71,456	3,913,920
512	287,167	286,272	31,035,776
1024	1,147,775	1,145,984	247,177,984
2048	4,589,311	4,585,728	1,972,979,200
4096	18,353,663	18,346,496	15,766,031,360

3.5.5 สรุปผลการวิเคราะห์

หากพิจารณาจากปริมาณของการสื่อสารและการคำนวณของวิธีการจัดรูปแบบข้อมูลทั้ง 3 รูปแบบข้างต้น พบว่า ในเรื่องของการสื่อสารวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกมีปริมาณของการสื่อสารน้อยที่สุด วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกเป็นอันดับ 2 และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกเป็นอันดับสุดท้าย โดยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกมีปริมาณการสื่อสารแตกต่างจาก 2 รูปแบบเป็นอย่างมาก เมื่อพิจารณาด้านปริมาณการคำนวณ พบว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกมีปริมาณการคำนวณน้อยเป็นอันดับหนึ่ง วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกเป็นอันดับ 2 และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกเป็นอันดับสุดท้าย การทำงานที่ก่อให้เกิดประสิทธิภาพนั้นต้องมีการสื่อสารและการคำนวณในปริมาณที่เหมาะสม หากพิจารณาตามข้อมูลที่ได้จากการวิเคราะห์ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ซึ่งเป็นวิธีการแบบใหม่ที่สร้างขึ้นในงานวิจัยชิ้นนี้ หากพิจารณาในเรื่องของความเร็วในการทำงานวิธีการใหม่นี้มีปริมาณการสื่อสารใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและมี

ปริมาณการคำนวณที่น้อยกว่า ส่วนวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกนั้นถึงแม้ว่าสามารถลดปริมาณการคำนวณลงได้ในปริมาณที่มากพอสมควร แต่ก็มีปริมาณการสื่อสารที่เพิ่มขึ้นมากเช่นกัน จึงน่าจะทำได้ดีกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไม่มากนัก เพราะฉะนั้นจึงตั้งสมมุติฐานว่า วิธีการแบบใหม่ที่สร้างขึ้น น่าจะทำงานได้มีเร็วกว่าทั้งสองวิธี แต่หากพิจารณาในเรื่องของความคุ้มค่าในการทำงาน วิธีการใหม่ที่สร้างขึ้นมีรูปแบบการทำงานคล้ายกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกจึงไม่ใช่วิธีการที่ใช้งานทรัพยากรได้อย่างคุ้มค่ามากนัก เนื่องจากเมื่อทำงานเสร็จในบางขั้นตอนมีหน่วยประมวลผลบางตัวที่ว่างงานลง

บทที่ 4

การทดลองและผลการทดลอง

บทที่ 4 การทดลองและผลการทดลอง เนื้อหาประกอบด้วย 2 ส่วน ส่วนที่ 1 การทดลอง นำเสนอวิธีที่ใช้ทดลองในงานวิจัย ส่วนที่ 2 ผลการทดลอง ได้นำเสนอถึงผลของการทดลองในแบบต่าง ๆ ของวิธีการจัดรูปแบบข้อมูล โดยได้นำเอาวิธีการจัดรูปแบบข้อมูลที่ได้วิเคราะห์และออกแบบไว้ในบทที่ 3 มาทำการทดลองด้วยการพัฒนาโปรแกรมประยุกต์ขึ้นมาเพื่อใช้ในการทดลอง ซึ่งโปรแกรมที่ได้ทำการพัฒนามานั้นได้นำมาใช้ในการทดลองเพื่อหา

1) ความเร็วในการทำงาน โดยวัดความเร็วในการทำงานจากเวลาที่ใช้ในการประมวลผลว่าในแต่ละวิธีการจัดรูปแบบข้อมูลใช้เวลาในการประมวลผลแตกต่างกันอย่างไร

2) อัตราความเร็วในการทำงานที่เพิ่มขึ้นเป็นอัตราความเร็วที่ได้จากการเปรียบเทียบเวลาที่ใช้ในการประมวลผลของการประมวลผลแบบลำดับและวิธีการจัดรูปแบบข้อมูลที่ประมวลผลแบบขนานแบบต่าง ๆ ที่ได้ออกแบบไว้ เพื่อให้ทราบว่าวิธีการจัดรูปแบบข้อมูลแบบขนานสามารถประมวลผลได้เร็วกว่าการประมวลผลแบบลำดับเป็นจำนวนกี่เท่า

4.1 การทดลอง

4.1.1 รูปแบบของการทดลอง

การทดลองนั้นแบ่งออกเป็นสองแบบ คือ การทำงานแบบลำดับและการทำงานแบบขนาน สำหรับการทำงานแบบขนานมีวิธีการจัดรูปแบบข้อมูล 3 วิธีด้วยกัน คือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก โดยแบ่งการทดลองออกเป็น 8 วิธีดังนี้

- 1) การประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถว
- 2) การประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบสลับแถว
- 3) การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
- 4) การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก
- 5) การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก

6) การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไจคลิก

7) การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไจคลิก

8) การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไจคลิก

ในส่วนของการทดลองได้ใช้ระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล โดยเมตริกซ์ A ที่นำมาใช้ในการทดลองนั้นต้องเป็นเมตริกซ์จัตุรัส สำหรับข้อมูลนำเข้าที่ใช้ในการทดลองนั้นเป็นข้อมูลที่ได้จากการสุ่มตัวเลขเมตริกซ์ที่ได้จากโปรแกรมที่พัฒนาขึ้นเพื่อใช้ในการสุ่มค่าตัวเลขเมตริกซ์ โดยแบ่งข้อมูลที่ใช้ในการทดลองออกเป็น 11 ชุด ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ชุดข้อมูลที่ใช้ในการทดลอง

ข้อมูลชุดที่	Matrix A	Vector B
1	4X4	4x1
2	8X8	8x1
3	16X16	16x1
4	32X32	32x1
5	64X64	64x1
6	128X128	128x1
7	256X256	256x1
8	512X512	512x1
9	1024X1024	1024x1
10	2048X2048	2048x1
11	4096X4096	4096x1

4.1.2 วิธีดำเนินการทดลอง

สำหรับการทดลองนี้มีวิธีการทำงานดังนี้

1) ทำการสุ่มค่าเมตริกซ์จากโปรแกรมที่สร้างขึ้น เพื่อให้ได้ข้อมูลที่เป็นเมตริกซ์ตามขนาดที่กำหนดไว้ในตารางที่ 4.1

2) ทดสอบการทำงานของโปรแกรมที่พัฒนาขึ้นทั้ง 8 รูปแบบ

3) จดบันทึกเวลาที่ใช้ในการทำงานของแต่ละรูปแบบ โดยในแต่ละรูปแบบได้ทำการทดลองหลายครั้ง จากนั้นจึงมาผลที่ได้มาหาค่าเฉลี่ยของเวลาที่ใช้ในการทำงาน

4) นำเวลาที่ได้จากวิธีการจัดรูปแบบข้อมูลแบบขนานมาหาค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้น

5) สร้างกราฟของเวลาที่ใช้ในการทำงานและอัตราความเร็วในการทำงานที่เพิ่มขึ้น

6) เปรียบเทียบประสิทธิภาพของการทำงานของวิธีการจัดรูปแบบข้อมูลแบบขนานว่ามีความเร็วในการทำงานมากกว่าการทำงานแบบลำดับหรือไม่

7) เปรียบเทียบประสิทธิภาพของการทำงานของแต่ละวิธีการจัดรูปแบบข้อมูลระหว่างการสลับแถวและไม่สลับแถวให้ประสิทธิภาพในการทำงานที่แตกต่างกันอย่างไร

8) เปรียบเทียบประสิทธิภาพของวิธีการจัดรูปแบบข้อมูลใหม่ที่ยกแบบไว้ว่า ทำงานได้มีประสิทธิภาพมากกว่าวิธีการจัดรูปแบบข้อมูลพื้นฐานทั้งสองแบบคือวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกหรือไม่

หมายเหตุ เนื่องจากในการนำเสนอผลการทดลองต้องมีการสร้างกราฟซึ่งมีความจำเป็นต้องใช้ชื่อของรูปแบบต่าง ๆ ดังนั้นจึงต้องมีการกำหนดความหมายของชื่อที่ใช้ดังนี้

1) gepp_s คือ โปรแกรมที่การประมวลผลแบบลำดับ ด้วยวิธีการกำจัดของเกาส์แบบสลับแถว

2) ge_s คือ โปรแกรมที่การประมวลผลแบบลำดับ ด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถว

3) gepp_c คือ โปรแกรมที่การประมวลผลแบบขนาน ด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

4) ge_c คือ โปรแกรมที่การประมวลผลแบบขนาน ด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

5) gepp_cbc คือ โปรแกรมที่การประมวลผลแบบขนาน ด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก

6) ge_cbc คือ โปรแกรมที่การประมวลผลแบบขนาน ด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก

7) gepp_cc คือ โปรแกรมที่การประมวลผลแบบขนาน ด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก

8) ge_cc คือ โปรแกรมที่การประมวลผลแบบขนาน ด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก

4.2 ผลการทดลอง

4.2.1 การประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถว

ผลการทดลองในเรื่องการใช้เวลาในการทำงานของการประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวข้อมูล โดยใช้ข้อมูลที่เป็นเมตริกซ์ขนาดต่าง ๆ กัน เวลาที่ใช้ในการประมวลผลแสดงในตารางที่ 4.2

ตารางที่ 4.2 เวลาที่ใช้ในการประมวลผลแบบลำดับของวิธีการกำจัดของเกาส์แบบไม่สลับแถว

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)
1	4×4	0.0000
2	8×8	0.0000
3	16×16	0.0000
4	32×32	0.0001
5	64×64	0.0008
6	128×128	0.0062
7	256×256	0.0490
8	512×512	0.4427
9	1024×1024	3.3900
10	2048×2048	25.8854
11	4096×4096	203.6002

จากตารางที่ 4.2 เป็นผลที่ได้จากการประมวลผลบนเครื่องที่ทำงานแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถว พบว่า ข้อมูลเมตริกซ์ขนาด 4×4 จนถึงขนาด 32×32 มีผลการทำงานที่ใกล้เคียงกันมากจนแทบจะ ไม่เห็นความแตกต่างของการใช้เวลาในการทำงาน หลังจากข้อมูลเมตริกซ์ขนาด 32×32 เป็นต้นไป จึงเห็นความแตกต่างของการใช้เวลาในการประมวลผลมากยิ่งขึ้น ช่วงของเมตริกซ์ที่มีขนาด 512×512 กับเมตริกซ์ที่มีขนาด 4096×4096 มีความแตกต่างของการใช้เวลาที่ชัดเจนมากที่สุด

4.2.2 การประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบสลับแถว

ผลการทดลองในเรื่องการใช้เวลาในการทำงานของการประมวลผลแบบลำดับโดยมีการสลับแถว ข้อมูลที่ใช้ในการทดลองเป็นเมตริกซ์ขนาดต่าง ๆ กันจำนวน 11 ชุด เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.3

ตารางที่ 4.3 เวลาที่ใช้ในการประมวลผลแบบลำดับของวิธีการกำจัดของเกาส์แบบสลับแถว

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)
1	4×4	0.0000
2	8×8	0.0000
3	16×16	0.0000
4	32×32	0.0002
5	64×64	0.0010
6	128×128	0.0067
7	256×256	0.0503
8	512×512	0.4583
9	1024×1024	3.4320
10	2048×2048	25.9241
11	4096×4096	204.7174

จากตารางที่ 4.3 เป็นผลที่ได้จากการประมวลผลบนเครื่องที่ทำงานแบบลำดับด้วยวิธีการกำจัดของเกาส์แบบมีการสลับแถวข้อมูล พบว่า ผลการใช้เวลาในการทำงานที่ได้มีค่าใกล้เคียงกับการทำงานแบบไม่สลับแถว แต่ใช้เวลาในการทำงานมากกว่าเล็กน้อยโดยข้อมูลเมทริกซ์ที่มีขนาด 4×4 จนถึง 32×32 ผลการทำงานมีลักษณะใกล้เคียงกันมากจนแทบไม่เห็นถึงความแตกต่าง หลังจากข้อมูลที่มีขนาด 32×32 เป็นต้นไป จึงเห็นความแตกต่างของการใช้เวลาในการประมวลผลมากยิ่งขึ้น ช่วงของข้อมูลที่มีขนาด 512×512 กับข้อมูลที่มีขนาด 4096×4096 นั้นสามารถเห็นความแตกต่างของการใช้เวลาที่ชัดเจนมากที่สุด

4.2.3 การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

ผลการทดลองในเรื่องการใช้เวลาในการประมวลผลแบบขนานด้วยวิธีการกำจัดแบบ เกาส์เขียนแบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกกับชุดข้อมูลที่เป็นเมทริกซ์ขนาดต่าง ๆ จำนวน 11 ชุด เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.4

ตารางที่ 4.4 เวลาที่ใช้ในการประมวลผลด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธี
การจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)
1	4×4	0.0007
2	8×8	0.0013
3	16×16	0.0031
4	32×32	0.0081
5	64×64	0.0180
6	128×128	0.0550
7	256×256	0.2083
8	512×512	0.8499
9	1024×1024	4.0775
10	2048×2048	19.3594
11	4096×4096	113.7938

จากตารางที่ 4.4 เป็นผลที่ได้จากการทำงานของวิธีการกำจัดของเกาส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์แบบไม่มีการสลับแถวข้อมูล พบว่า ข้อมูลที่มีขนาด 4×4 จนถึงขนาด 512×512 ใช้เวลาในการทำงานน้อยกว่า 1 วินาทีและยิ่งเมตริกซ์มีขนาดใหญ่ขึ้นเท่าใด การใช้เวลาในการประมวลผลก็ยิ่งเพิ่มมากขึ้น โดยข้อมูลที่มีขนาด 2048×2048 ถึง 4096×4096 มีช่วงห่างของผลการทำงานที่ชัดเจนที่สุด หากเปรียบเทียบกับ การประมวลผลแบบลำดับข้อมูลที่มีขนาด 2048×2048 เป็นต้นไปของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกสามารถทำงานได้เร็วกว่าประมวลผลแบบลำดับ

4.2.4 การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก

ผลการทดลองในเรื่องการใช้เวลาในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกกับชุดข้อมูลที่เป็นเมตริกซ์ขนาดต่าง ๆ จำนวน 11 ชุด เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.5

ตารางที่ 4.5 เวลาที่ใช้ในการประมวลผลด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกบนระบบคลัสเตอร์ขนาด 4 โหนด

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)
1	4×4	0.0008
2	8×8	0.0014
3	16×16	0.0033
4	32×32	0.0086
5	64×64	0.0188
6	128×128	0.0568
7	256×256	0.2105
8	512×512	0.8562
9	1024×1024	4.1669
10	2048×2048	19.4606
11	4096×4096	114.6023

จากตารางที่ 4.5 เป็นผลที่ได้จากการทำงานของวิธีการกำจัดของเกาส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกแบบมีการสลับแถวข้อมูล พบว่า เวลาในการทำงานมีค่าใกล้เคียงกับการทำงานแบบไม่สลับแถวข้อมูลยิ่งเมตริกซ์มีขนาดใหญ่ขึ้นเท่าใด การใช้เวลาในการประมวลผลก็ยิ่งเพิ่มมากขึ้น โดยเมตริกซ์ที่มีขนาด 4×4 จนถึงขนาด 512×512 ใช้เวลาในการประมวลผลน้อยกว่า 1 วินาที การใช้เวลาในการทำงานมีทิศทางการเพิ่มเช่นเดียวกันกับแบบที่ไม่มีการสลับแถวข้อมูลแต่ใช้เวลาในการทำงานมากกว่า

4.2.5 การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก

ผลการทดลองในเรื่องการใช้เวลาในการทำงานของการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกกับชุดข้อมูลที่เป็นเมตริกซ์ขนาดต่าง ๆ จำนวน 9 ชุด ตั้งแต่ขนาด 16×16 จนถึง 4096×4096 โดยแบ่งข้อมูลออกเป็นบล็อกข้อมูลที่มีขนาดต่างกัน 8 กลุ่มคือ 4, 8, 16, 32, 64, 128, 256 และ 512 เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.6

ตารางที่ 4.6 เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับ
แถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก ทำงานบนระบบ
คลัสเตอร์ขนาด 4 หน่วยประมวลผล

ข้อมูล	ขนาดบล็อก							
	4	8	16	32	64	128	256	512
16	0.0030							
32	0.0843	0.0080						
64	0.3601	0.1515	0.0179					
128	1.2451	0.6032	0.2032	0.0522				
256	2.2884	1.3469	0.8403	0.5785	0.2030			
512	5.3922	3.3477	1.8035	1.3293	1.1504	0.8405		
1024	12.9374	7.9881	5.9329	5.4934	5.1879	4.8551	4.0680	
2048	35.5294	27.5714	24.6282	24.2777	23.5373	22.9686	21.6473	19.3348
4096	135.9161	123.7225	121.3966	120.1616	119.1505	118.2589	116.2257	113.5147

จากตารางที่ 4.6 เป็นผลที่ได้จากการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก พบว่า ยิ่งเมื่อมีการแบ่งขนาดข้อมูลเป็นกลุ่มที่ใหญ่ขึ้นเรื่อย ๆ ทำให้สามารถทำงานได้เร็วขึ้น โดยทดลองทำงานบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล ข้อมูลเมตริกซ์เริ่มต้นขนาด 16×16 จนถึง 4096×4096 แบ่งบล็อกข้อมูลในการทดสอบออกเป็น 8 กลุ่มในการทำงานบล็อกข้อมูลที่มีขนาด 512 ทำงานกับชุดข้อมูลที่มีขนาด 2048×2048 และ 4096×4096 ใช้เวลาในการทำงานน้อยที่สุด

4.2.6 การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก

ผลการทดลองในเรื่องการใช้เวลาในการทำงานของการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกกับชุดข้อมูลที่เป็นเมตริกซ์ขนาดต่าง ๆ จำนวน 9 ชุด ตั้งแต่ขนาด 16×16 จนถึง 4096×4096 โดยแบ่งข้อมูลออกเป็นบล็อกข้อมูลที่มีขนาดต่างกัน 8 กลุ่มคือ 4, 8, 16, 32, 64, 128, 256 และ 512 เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.7

ตารางที่ 4.7 เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถว
โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก บนระบบคลัสเตอร์
ขนาด 4 หน่วยประมวลผล

ข้อมูล	ขนาดบล็อก							
	4	8	16	32	64	128	256	512
16	0.0031							
32	0.0854	0.0082						
64	0.3877	0.1556	0.0185					
128	1.2972	0.6612	0.2098	0.0547				
256	2.4360	1.6037	0.8540	0.5900	0.2089			
512	5.5770	3.3646	1.9880	1.3383	1.3604	0.8513		
1024	13.0622	8.3188	6.3294	5.5288	5.2002	4.9422	4.1101	
2048	36.2883	27.6392	24.9483	24.4511	24.3954	23.2511	21.8745	19.4211
4096	136.5369	124.9681	122.4336	121.5093	120.5955	119.2028	117.1045	114.4942

จากตารางที่ 4.7 เป็นผลที่ได้จากการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก พบว่า เวลาที่ใช้ในการทำงานมีค่าที่ใกล้เคียงกับการทำงานแบบไม่มีการสลับแถวแต่ใช้เวลาในการทำงานมากกว่า เมื่อมีการแบ่งขนาดข้อมูลเป็นกลุ่มที่ใหญ่ขึ้นเรื่อย ๆ ทำให้สามารถทำงานได้เร็วขึ้น โดยทดลองทำงานบนระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล ข้อมูลเมตริกซ์เริ่มต้นขนาด 16×16 จนถึง 4096×4096 แบ่งบล็อกข้อมูลในการทดสอบออกเป็น 8 กลุ่มในการทำงานบล็อกข้อมูลที่มีขนาด 512 ซึ่งเป็นขนาดบล็อกที่ใหญ่ที่สุดในการทดลองครั้งนี้ทำงานกับชุดข้อมูลที่มีขนาด 2048×2048 และ 4096×4096 ใช้เวลาในการทำงานน้อยที่สุด

4.2.7 การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก

ผลการทดลองในเรื่องการใช้เวลาในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกทำการทดลองโดยใช้ข้อมูลที่เป็นเมตริกซ์ขนาดต่างกันจำนวน 11 ชุดข้อมูล เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.8

ตารางที่ 4.8 เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับ
แถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกบน
ระบบคลัสเตอร์ขนาด 4 หน่วยประมวลผล

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)
1	4×4	0.0006
2	8×8	0.0012
3	16×16	0.0029
4	32×32	0.0074
5	64×64	0.0174
6	128×128	0.0527
7	256×256	0.1970
8	512×512	0.8009
9	1024×1024	3.5921
10	2048×2048	17.4498
11	4096×4096	98.5529

จากตารางที่ 4.8 เป็นผลที่ได้จากการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบไม่สลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก พบว่า ผลการทดลองมีลักษณะที่ใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก แต่ใช้เวลาในการทำงานน้อยกว่าเล็กน้อย โดยเมตริกซ์ที่มีขนาด 4×4 จนถึงขนาด 512×512 ใช้เวลาในการประมวลผลน้อยกว่า 1 วินาที เมื่อนำเอาวิธีการนี้ไปเปรียบเทียบกับผลการประมวลผลแบบลำดับข้อมูลที่มีขนาด 2048×2048 เป็นต้นไปสามารถทำงานได้เร็วกว่าการประมวลผลแบบลำดับ

4.2.8 การประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถวโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกแบบสลับแถว

ผลการทดลองในเรื่องการใช้เวลาในการทำงานของการประมวลผลแบบขนานโดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกแบบมีการสลับแถวข้อมูล ทำการทดลองโดยใช้ข้อมูลที่เป็นเมตริกซ์ขนาดต่างกันจำนวน 11 ชุดข้อมูล เวลาที่ใช้ในการทำงานแสดงในตารางที่ 4.9

ตารางที่ 4.9 เวลาที่ใช้ในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์แบบสลับแถว
โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกบนระบบ
คลัสเตอร์ขนาด 4 หน่วยประมวลผล

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)
1	4×4	0.0007
2	8×8	0.0013
3	16×16	0.0030
4	32×32	0.0079
5	64×64	0.0178
6	128×128	0.0539
7	256×256	0.2006
8	512×512	0.8051
9	1024×1024	3.8646
10	2048×2048	17.5639
11	4096×4096	98.6022

จากตารางที่ 4.9 เป็นผลที่ได้จากการทำงานของวิธีการกำจัดของเกาส์ที่ทำงานบนเครื่องที่ทำงานแบบขนาน โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกแบบมีการสลับแถวข้อมูล พบว่า ผลการทดลองมีลักษณะที่ใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก แต่ใช้เวลาในการทำงานน้อยกว่าเล็กน้อย โดยเมตริกซ์ที่มีขนาด 4×4 จนถึงขนาด 512×512 ใช้เวลาในการประมวลผลน้อยกว่า 1 วินาที เมื่อนำเอาวิธีการนี้ไปเปรียบเทียบกับวิธีการประมวลผลแบบลำดับข้อมูลที่มีขนาด 2048×2048 เป็นต้นไปสามารถทำงานได้เร็วกว่าการประมวลผลแบบลำดับ

4.3 การเปรียบเทียบประสิทธิภาพในการทำงาน

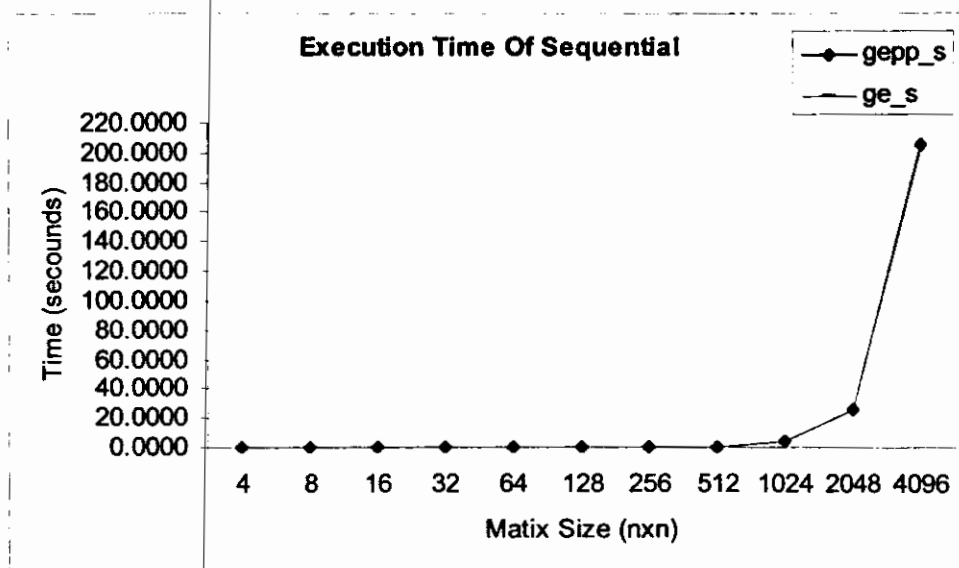
4.3.1 การเปรียบเทียบเวลาในการประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์ระหว่างแบบไม่สลับแถวและแบบสลับแถว

การเปรียบเทียบเวลาในการประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์ระหว่างแบบไม่สลับแถวและแบบสลับแถว ผลปรากฏว่าแบบ ไม่มีการสลับแถวข้อมูลใช้เวลาในการทำงานน้อยกว่าแบบมีการสลับแถวข้อมูล ซึ่งผลต่างของการใช้เวลาในการทำงานระหว่างทั้งสองวิธีห่างกันไม่มากนัก ดังแสดงในตารางที่ 4.10

ตารางที่ 4.10 การเปรียบเทียบเวลาในการประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์ระหว่างแบบไม่สลับแถวและแบบสลับแถว

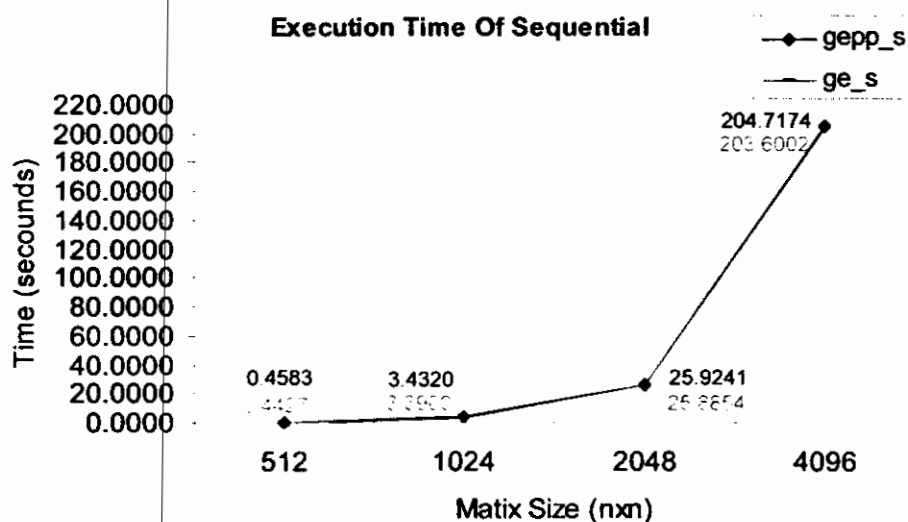
ข้อมูลชุดที่	ขนาด	เวลาที่ใช้ในการประมวลผล (วินาที)		ระยะห่างเวลา (วินาที)
		มีการสลับแถว	ไม่มีการสลับแถว	
1	4×4	0.0000	0.0000	0.0000
2	8×8	0.0000	0.0000	0.0000
3	16×16	0.0000	0.0000	0.0000
4	32×32	0.0002	0.0001	0.0001
5	64×64	0.0010	0.0008	0.0002
6	128×128	0.0067	0.0062	0.0005
7	256×256	0.0503	0.0490	0.0013
8	512×512	0.4583	0.4427	0.0156
9	1024×1024	3.4320	3.3900	0.0420
10	2048×2048	25.9241	25.8854	0.0387
11	4096×4096	204.7174	203.6002	1.1172

จากตารางที่ 4.10 พบว่า เมตริกซ์ที่มีขนาด 32×32 เป็นต้นไปของการทำงานแบบ ไม่มีการสลับแถวสามารถทำงานได้เร็วกว่าการทำงานแบบมีการสลับแถว โดยเวลาที่ใช้ในการประมวลผลของทั้งสองแบบได้เปรียบเทียบให้เห็นดังแสดงในภาพที่ 4.1 จากกราฟที่ได้ จะเห็นว่าการใช้เวลาในการทำงานมีค่าใกล้เคียงกันมาก โดยการสลับแถวข้อมูลใช้เวลาในการทำงานมากกว่าเล็กน้อย



ภาพที่ 4.1 กราฟแสดงการเปรียบเทียบเวลาในการประมวลผลแบบลำดับด้วยวิธีการกำจัดของเกาส์ระหว่างแบบไม่สลับแถวและแบบสลับแถว

เนื่องจากช่วงข้อมูลของกราฟห่างกันจึงทำให้กราฟในภาพที่ 4.1 ดูไม่ชัดเจน จึงได้คัดเอาบางช่วงของข้อมูลมาแสดงเป็นกราฟดังแสดงในภาพที่ 4.2 รูปแบบที่มีการสลับแถวและไม่มีสลับแถวใช้เวลาต่างกันเล็กน้อย โดยรูปแบบที่ใช้การสลับแถวข้อมูลทำงานได้ช้ากว่าเล็กน้อย



ภาพที่ 4.2 กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลแบบลำดับเมตริกซ์ขนาด 512×512 ถึง 4069×4096 ทั้งแบบมีการสลับแถวข้อมูลและแบบไม่มีการสลับแถว

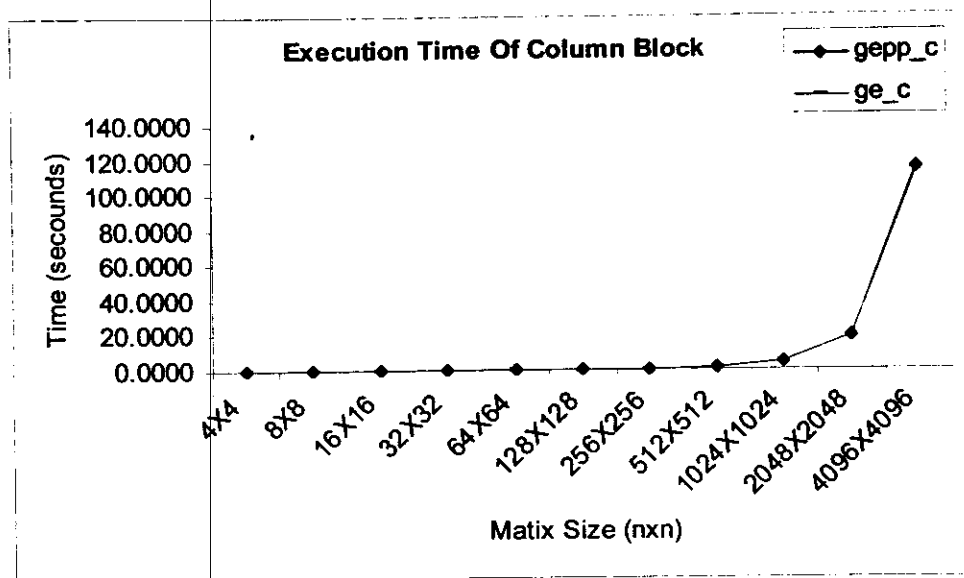
4.3.2 การเปรียบเทียบเวลาในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกระหว่างแบบไม่สลับแถวและแบบสลับแถว

การเปรียบเทียบความเร็วในการประมวลผลของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกระหว่างแบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล ผลปรากฏว่าแบบไม่มีการสลับแถวข้อมูลใช้เวลาในการทำงานน้อยกว่าแบบมีการสลับแถวข้อมูลในทุกชุดข้อมูลที่ทำงานทดลอง ดังแสดงในตารางที่ 4.11

ตารางที่ 4.11 การเปรียบเทียบเวลาในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกระหว่างแบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล

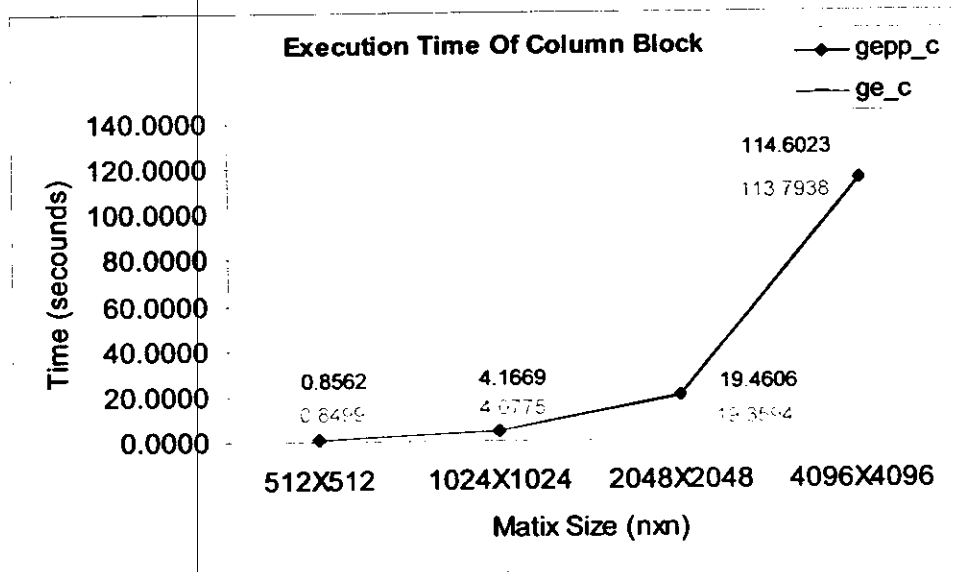
ข้อมูลชุดที่	ขนาด	เวลาที่ใช้ในการประมวลผล (วินาที)		ระยะห่างเวลา (วินาที)
		มีการสลับแถว	ไม่มีการสลับแถว	
1	4×4	0.0008	0.0007	0.0001
2	8×8	0.0014	0.0013	0.0001
3	16×16	0.0033	0.0031	0.0002
4	32×32	0.0086	0.0081	0.0005
5	64×64	0.0188	0.0180	0.0008
6	128×128	0.0568	0.0550	0.0018
7	256×256	0.2105	0.2083	0.0022
8	512×512	0.8562	0.8499	0.0063
9	1024×1024	4.1669	4.0775	0.0894
10	2048×2048	19.4606	19.3594	0.1012
11	4096×4096	114.6023	113.7938	0.8085

จากตารางที่ 4.11 เป็นการเปรียบเทียบผลที่ได้จากการทำงานของวิธีการกำจัดของเกาส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก แบบไม่มีการสลับแถวข้อมูลสามารถทำงานได้เร็วกว่าแบบมีการสลับแถวข้อมูลเล็กน้อย ผลการเปรียบเทียบเวลาที่ใช้ในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกทั้งสองแบบแสดงให้เห็นในกราฟภาพที่ 4.3



ภาพที่ 4.3 กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ระหว่างแบบมีการสลับแถวข้อมูลและ ไม่มีการสลับแถวข้อมูล

เนื่องจากผลการทดลองที่ได้มีความใกล้เคียงกันมาก เพราะฉะนั้นเพื่อให้เห็นข้อมูลได้ชัดเจนขึ้นจึงได้คัดเลือกเฉพาะข้อมูลบางส่วนของการทำงานด้วยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกมาทำเป็นกราฟใหม่ ดังกราฟในภาพที่ 4.4 ทั้งรูปแบบมีการสลับแถวข้อมูลและรูปแบบที่ไม่มีการสลับแถวข้อมูลใช้เวลาในการทำงานไม่แตกต่างกันมากนัก



ภาพที่ 4.4 กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก ช่วงข้อมูล 512 ถึง 4096 ระหว่างแบบมีการสลับแถวและแบบไม่มีการสลับแถว

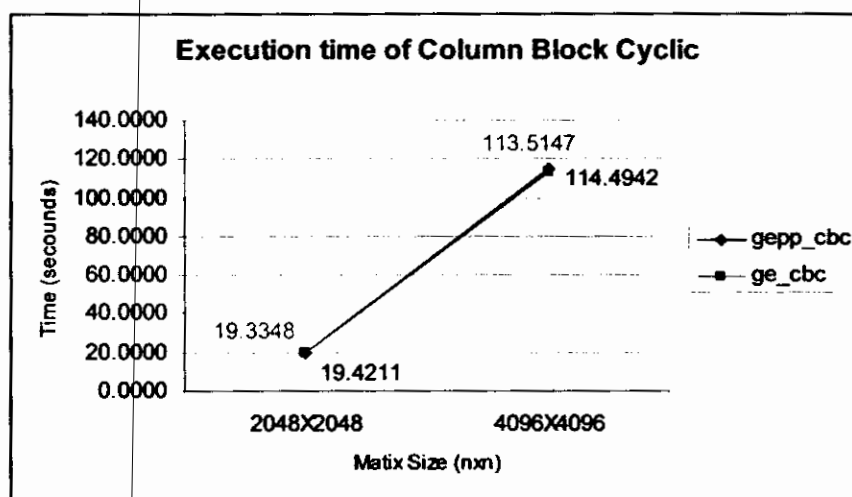
4.3.3 การเปรียบเทียบเวลาในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกสส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกระหว่างแบบไม่สลับแถวและแบบสลับแถว

การเปรียบเทียบความเร็วในการประมวลผลของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกระหว่างแบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล ผลปรากฏว่าแบบไม่มีการสลับแถวข้อมูลใช้เวลาในการทำงานน้อยกว่าแบบมีการสลับแถวข้อมูลในทุกชุดข้อมูลที่ทำงานทดลอง โดยได้นำเอาชุดข้อมูลที่มีผลการทำงานดีที่สุดคือ บล็อกข้อมูลขนาด 512 มาแสดงในตารางที่ 4.12

ตารางที่ 4.12 การเปรียบเทียบเวลาในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกระหว่างแบบที่ไม่มีการสลับแถวข้อมูลและแบบที่มีการสลับแถวข้อมูล บล็อกข้อมูลขนาด 512

ข้อมูลชุดที่	ขนาดของข้อมูล	เวลาที่ใช้ในการประมวลผล (วินาที)		เวลาที่ต่าง
		มีการสลับแถว	ไม่มีการสลับแถว	
1	2048X2048	19.4211	19.3348	0.0863
2	4096X4096	114.4942	113.5147	0.9795

จากตารางที่ 4.12 รูปแบบการทำงานแบบไม่มีการสลับแถวข้อมูลทำงานได้เร็วกว่าแบบมีการสลับแถวข้อมูลเล็กน้อย ดังแสดงในรูปที่ 4.5



ภาพที่ 4.5 กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกระหว่างแบบมีการสลับแถวข้อมูลและไม่มีการสลับแถวข้อมูล

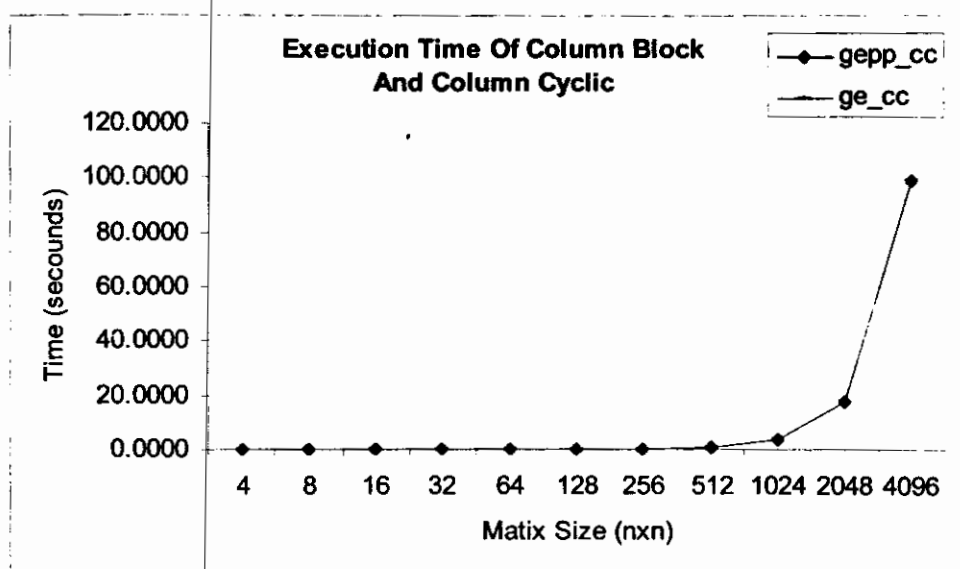
4.3.4 การเปรียบเทียบเวลาในการประมวลผลแบบขนานด้วยวิธีการกำจัดของเกาส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างแบบไม่สลับแถวและแบบสลับแถว

การเปรียบเทียบความเร็วในการประมวลผลของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างแบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล ผลปรากฏว่าแบบไม่มีการสลับแถวข้อมูลใช้เวลาในการทำงานน้อยกว่าแบบมีการสลับแถวข้อมูลในทุกชุดข้อมูลที่ทำงานทดลอง ดังแสดงในตารางที่ 4.13

ตารางที่ 4.13 การเปรียบเทียบเวลาในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างแบบไม่มีการสลับแถวและแบบมีการสลับแถว

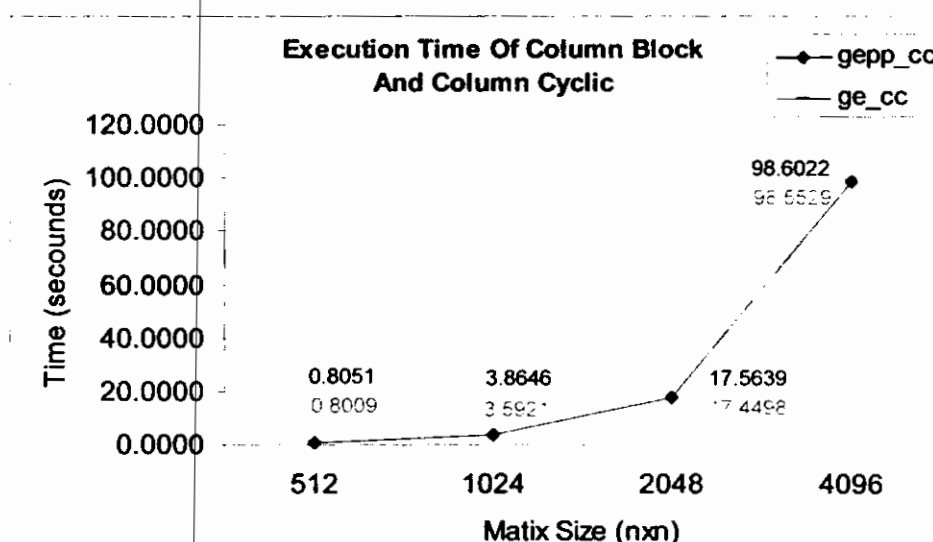
ข้อมูลชุดที่	ขนาด	เวลาที่ใช้ในการประมวลผล (วินาที)		ระยะห่างเวลา (วินาที)
		มีการสลับแถว	ไม่มีการสลับแถว	
1	4×4	0.0007	0.0006	0.0001
2	8×8	0.0013	0.0012	0.0001
3	16×16	0.0030	0.0029	0.0001
4	32×32	0.0079	0.0074	0.0005
5	64×64	0.0178	0.0174	0.0004
6	128×128	0.0539	0.0527	0.0012
7	256×256	0.2006	0.1970	0.0036
8	512×512	0.8051	0.8009	0.0042
9	1024×1024	3.8646	3.5921	0.2725
10	2048×2048	17.5639	17.4498	0.1141
11	4096×4096	98.6022	98.5529	0.0493

จากตารางที่ 4.13 เป็นการเปรียบเทียบผลที่ได้จากการทำงานของวิธีการกำจัดของเกาส์โดยใช้วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกแบบไม่มีการสลับแถวข้อมูล สามารถทำงานได้เร็วกว่าแบบมีการสลับแถวข้อมูลเล็กน้อย ผลการเปรียบเทียบเวลาที่ใช้ในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างทั้งสองแบบแสดงให้เห็นในกราฟภาพที่ 4.6



ภาพที่ 4.6 กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลผลวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างแบบมีการสลับแถวและไม่มีการสลับแถว

เนื่องจากผลการทดลองที่ได้มีความใกล้เคียงกันมาก เพราะฉะนั้นเพื่อให้เห็นข้อมูลได้ชัดเจนขึ้นจึงได้ตัดเอาเฉพาะข้อมูลบางส่วนของการทำงานด้วยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกมาทำเป็นกราฟใหม่ดังกราฟในภาพที่ 4.7



ภาพที่ 4.7 กราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการทำงานของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกระหว่างแบบสลับแถวและแบบไม่มีการสลับแถว

4.3.5 อัตราความเร็วในการทำงานที่เพิ่มขึ้น

อัตราความเร็วในการทำงานที่เพิ่มขึ้น เป็นอัตราความเร็วที่ได้จากการเปรียบเทียบ เวลาที่ใช้ในการประมวลผลของการประมวลผลแบบลำดับและวิธีการจัดรูปแบบข้อมูล ที่ประมวลผลแบบขนาน Karp and Falit [23] ได้กล่าวไว้ในงานวิจัยว่า อัตราความเร็วในการทำงานที่ เพิ่มขึ้นเป็นค่าที่บ่งบอกว่าโปรแกรมที่ประมวลผลแบบขนาน สามารถประมวลผลได้เร็วกว่า โปรแกรมที่ประมวลผลแบบลำดับเป็นจำนวนกี่เท่า โดยการหาค่าอัตราความเร็วในการทำงานที่ เพิ่มขึ้น เมื่อนำเอาเวลาที่ใช้ในการทำงานของการประมวลผลแบบลำดับและเวลาที่ใช้ในการทำงาน ของการประมวลผลแบบขนานมาหารหาค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้นนั้นหาได้จากสูตร ดังนี้

$$Speedup = \frac{S_i}{P_i} \quad (2)$$

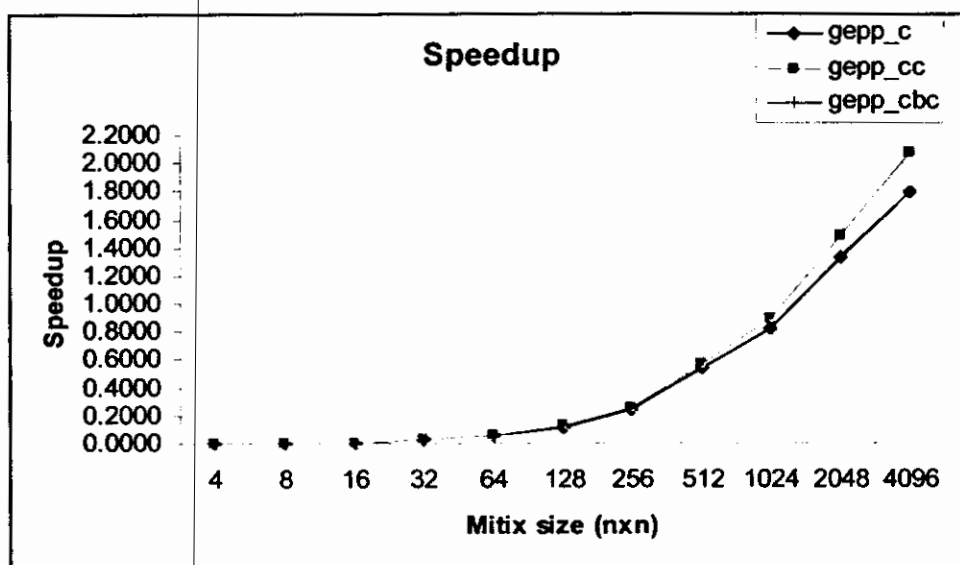
เมื่อ S_i คือ เวลาที่ใช้ในแต่ละชุดข้อมูลของการทำงานแบบลำดับ
 P_i คือ เวลาที่ใช้ในแต่ละชุดข้อมูลของการทำงานแบบขนาน

เมื่อนำเอาผลที่ได้จากการทดลองมาแทนค่าในสูตร ได้ผลดังตารางที่ 4.11

ตารางที่ 4.14 เปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นในการทำงานของแต่ละวิธีแบบมี การสลับแถวข้อมูล

ชุด	ขนาด	gepp_c	gepp_cc	gepp_cbc
1	4	0.0000	0.0000	
2	8	0.0000	0.0000	
3	16	0.0000	0.0000	
4	32	0.0233	0.0253	
5	64	0.0532	0.0562	
6	128	0.1180	0.1243	
7	256	0.2390	0.2507	
8	512	0.5353	0.5692	
9	1024	0.8236	0.8881	
10	2048	1.3321	1.4760	1.3348
11	4096	1.7863	2.0762	1.7880

จากตารางที่ 4.14 เป็นผลที่ได้จากการหาค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้น หากเปรียบเทียบระหว่างวิธีการจัดรูปแบบข้อมูลระหว่างวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก โดยทั้งสามแบบทำงานแบบมีการสลับแถวข้อมูล ในส่วนของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกซึ่งมีการแบ่งการทดลองออกเป็นหลายกลุ่ม โดยในแต่ละกลุ่มมีขนาดของบล็อกที่แตกต่างกัน จึงได้นำเอาผลการทำงานที่ดีที่สุดมาทำการเปรียบเทียบ คือ บล็อกที่มีขนาด 512 จากการเปรียบเทียบพบว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกซึ่งเป็นวิธีการใหม่ที่ยกแบบขึ้นในการทำวิจัยครั้งนี้มีค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้นสูงกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก ดังแสดงในภาพที่ 4.8



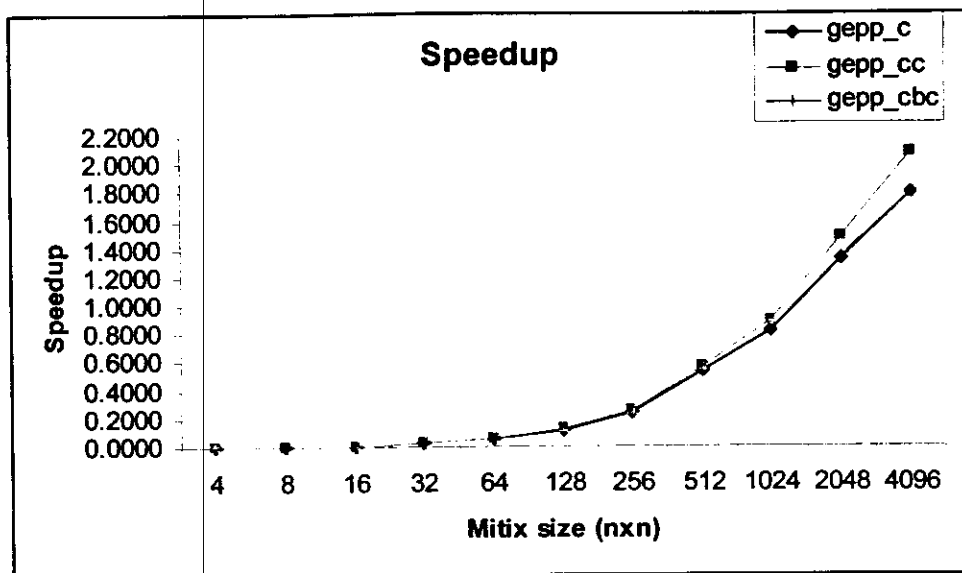
ภาพที่ 4.8 กราฟแสดงการเปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิก และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิก ทำงานแบบมีการสลับแถว

จากกราฟในภาพที่ 4.8 วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไขว้คลิกมีอัตราความเร็วในการทำงานที่เพิ่มขึ้นมากกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไขว้คลิก โดยเมื่อข้อมูลมีขนาดไม่ใหญ่มากจะยังไม่เห็นความแตกต่างของอัตราความเร็วในการทำงานมากนักแต่เมื่อข้อมูลมีขนาดที่ใหญ่ขึ้นจะเห็นความแตกต่างที่ชัดเจนมากขึ้น สำหรับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไขว้คลิกสามารถทำงานได้ดีกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกเล็กน้อย

ตารางที่ 4.15 เปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นในการทำงานของแต่ละวิธีแบบไม่มี
การสลับแถวข้อมูล

ชุด	ขนาด	ge_c	ge_cc	ge_cbc
1	4	0.0000	0.0000	
2	8	0.0000	0.0000	
3	16	0.0000	0.0000	
4	32	0.0123	0.0135	
5	64	0.0444	0.0460	
6	128	0.1127	0.1176	
7	256	0.2352	0.2487	
8	512	0.5209	0.5528	
9	1024	0.8314	0.9437	
10	2048	1.3371	1.4834	1.3388
11	4096	1.7892	2.0659	1.7936

จากตารางที่ 4.15 เป็นผลที่ได้จากการหาค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้น หากเปรียบเทียบระหว่างวิธีการจัดรูปแบบข้อมูลระหว่างวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก โดยทั้งสามแบบทำงานแบบไม่มีการสลับแถวข้อมูล ในส่วนของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกซึ่งมีการแบ่งการทดลองออกเป็นหลายกลุ่มโดยในแต่ละกลุ่มมีขนาดของบล็อกที่แตกต่างกัน จึงได้นำเอาผลการทำงานที่ดีที่สุดมาทำการเปรียบเทียบขนาดบล็อกที่ทำงานได้ดีที่สุดที่ได้จากการทดลองของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก คือ 512 จากการเปรียบเทียบพบว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ซึ่งเป็นวิธีการใหม่ที่ออกแบบขึ้นในการทำวิจัยครั้งนี้มีค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้นสูงกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก ดังแสดงในภาพที่ 4.9



ภาพที่ 4.9 กราฟแสดงการเปรียบเทียบอัตราความเร็วในการทำงานที่เพิ่มขึ้นของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ทำงานแบบไม่มีการสลับแถว

จากกราฟในภาพที่ 4.8 และ 4.9 พบว่า วิธีการจัดรูปแบบข้อมูลแบบใหม่ คือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกนั้นมีค่าอัตราความเร็วในการทำงานมากกว่าวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและแบบคอลัมน์บล็อกไซคลิกทั้งแบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล ซึ่งเป็นไปตามสมมุติฐานที่ได้ออกแบบเอาไว้ในบทที่ 3 โดยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกแบบไม่มีการสลับแถวมีค่าอัตราความเร็วในการทำงานที่เพิ่มขึ้นมากที่สุดตั้งแต่ชุดข้อมูลขนาด 32×32 จนถึง 4096×4096

บทที่ 5

สรุปผล

ในบทนี้ เนื้อหาประกอบด้วยข้อมูล 2 ส่วน ส่วนที่ 1 สรุปผลการศึกษา ซึ่งเป็นส่วนที่ได้ทำการสรุปผลที่ได้จากการวิจัย และส่วนที่ 2 คือข้อเสนอแนะ เพื่อเป็นแนวทางในการทำงานวิจัย ต่อเนื่องจากงานวิจัยนี้

5.1 สรุปผลการศึกษา

การวิจัยนี้ ได้ศึกษาการนำเอาวิธีการกำจัดของเกาส์มาประมวลผลแบบขนานบนระบบคลัสเตอร์ จุดหลักของการศึกษาก็คือวิธีการจัดรูปแบบข้อมูล ซึ่งเป็นพื้นฐานสำคัญต่อประสิทธิภาพของการประมวลผลแบบขนาน หากจะนำเอาวิธีการจัดรูปแบบข้อมูลมาทำงานโดยใช้วิธีการกำจัดของเกาส์ จะต้องหาวิธีการจัดรูปแบบที่เหมาะสมกับวิธีการนี้ เนื่องจากกระบวนการในการกำจัดค่าตัวแปรด้วยวิธีการกำจัดของเกาส์นั้นมีขั้นตอนของการสลับแถวข้อมูลเพื่อลดข้อผิดพลาดจากค่าในแนวเส้นทแยงมุมที่นำมาหารเป็น 0 การสลับแถวนั้นจะทำการสลับแถวที่อยู่ในคอลัมน์เดียวกัน จึงจำเป็นต้องใช้วิธีการจัดรูปแบบข้อมูลที่มีลักษณะการทำงานแบบคอลัมน์เพราะว่า ถ้าหากใช้วิธีการจัดรูปแบบข้อมูลแบบอื่นจะทำให้เกิดการสื่อสารเกิดขึ้นในปริมาณที่มาก ซึ่งมีผลต่อประสิทธิภาพในการทำงาน สำหรับวิธีการจัดรูปแบบข้อมูลที่มีลักษณะการทำงานแบบคอลัมน์นั้นมีหลายวิธีด้วยกันคือ วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก, วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก โดยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกมีปัญหาในเรื่องปริมาณการคำนวณที่เกิดขึ้นมากและการใช้งานทรัพยากรได้ไม่คุ้มค่า ส่วนวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิกเป็นวิธีที่พัฒนามาจากวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกโดยพยายามลดปริมาณการคำนวณลงและใช้งานทรัพยากรให้คุ้มค่ามากขึ้นและเป็นวิธีที่ทำงานได้มีประสิทธิภาพที่มากที่สุดเมื่อเปรียบเทียบกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ทั้งสามวิธีแต่ก็มีปัญหาในเรื่องของปริมาณการสื่อสารที่เกิดขึ้น จึงไม่ได้ประสิทธิภาพเพิ่มมากขึ้นเท่าที่ควร ดังนั้นหากมีวิธีการจัดรูปแบบข้อมูลที่มีปริมาณการสื่อสารใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและมีปริมาณการคำนวณใกล้เคียงกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก และวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกน่าจะทำงานได้ดีกว่าทั้งสามวิธี จึงได้ออกแบบวิธีการจัดรูปแบบข้อมูล เพื่อใช้ในการแก้ปัญหาระบบสมการเชิงเส้นด้วยวิธีการกำจัดของเกาส์ที่ทำ

งานแบบขนานบนระบบเครือข่ายคลัสเตอร์รุ่นใหม่ โดยอาศัยวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิกเป็นต้นแบบเพื่อต้องการให้ได้วิธีการจัดรูปแบบข้อมูลที่ทำงานได้อย่างมีประสิทธิภาพมากขึ้นและลดข้อบกพร่องของวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์ไซคลิก

ในส่วนของ การทดลอง ได้ใช้เครือข่ายคลัสเตอร์ขนาด 4 หน่วยประมวลผล ข้อมูลที่นำมาทดลองเป็นเมตริกซ์ โดยเมตริกซ์ A ที่นำมาใช้ในการทดลองนั้นต้องเป็นเมตริกซ์แบบจตุรัส โดยได้แบ่งข้อมูลที่นำมาใช้ในการทดลองออกเป็น 11 ชุด คือ 4×4 , 8×8 , 16×16 , 32×32 , 64×64 , 128×128 , 256×256 , 512×512 , 1024×1024 , 2048×2048 , 4096×4096

การทดสอบประสิทธิภาพนั้นได้ใช้เวลาที่ใช้ในการประมวลผล และอัตราความเร็วในการทำงานที่เพิ่มขึ้นมาใช้เป็นปัจจัยวัดประสิทธิภาพในการทำงาน นอกจากนั้นในแต่ละวิธีการจัดรูปแบบข้อมูลได้เปรียบเทียบการทำงาน 2 แบบ คือ แบบไม่มีการสลับแถวข้อมูลและแบบมีการสลับแถวข้อมูล ว่ามีประสิทธิภาพในการทำงานแตกต่างกันอย่างไร การทดลองงานวิจัยนี้ ได้พัฒนาโปรแกรมที่ประมวลผลแบบขนาน เพื่อใช้แก้ปัญหасสมการเชิงเส้น $Ax=b$ โดยโปรแกรมที่พัฒนาขึ้นใช้วิธีการกำจัดของเกาส์ในการแก้ปัญหасสมการเชิงเส้น

ผลการทดลองประสิทธิภาพของวิธีการกระจายข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก ซึ่งเป็นวิธีการจัดรูปแบบข้อมูลแบบใหม่ที่ได้สร้างขึ้นในการวิจัยครั้งนี้ เมื่อนำเอาเวลาที่ใช้ในการทำงานเมื่อนำมาเปรียบเทียบกับการประมวลผลแบบลำดับ พบว่า หากเมตริกซ์มีขนาดเล็กการประมวลผลแบบลำดับใช้เวลาในการทำงานน้อยกว่า แต่เมื่อเมตริกซ์มีขนาดที่ใหญ่ขึ้น (2048×2048) วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิก สามารถทำงานได้อย่างมีประสิทธิภาพมากกว่า เนื่องจากใช้เวลาในการทำงานน้อยกว่า เมื่อนำมาเปรียบเทียบกับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไซคลิก ซึ่งเป็นรูปแบบการทำงานบนระบบขนานเหมือนกัน พบว่า ใช้เวลาในการทำงานน้อยกว่าทุกชุดข้อมูล เมื่อนำเอาวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกมาเปรียบเทียบกับส่วนของอัตราความเร็วที่เพิ่มขึ้นระหว่างวิธีการจัดรูปแบบข้อมูลทั้ง 3 รูปแบบ พบว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกมีอัตราความเร็วในการทำงานที่ดีที่สุด จึงสรุปได้ว่า วิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและคอลัมน์ไซคลิกซึ่งเป็นวิธีการกระจายข้อมูลแบบใหม่ที่สร้างขึ้นมีประสิทธิภาพมากกว่าวิธีการต้นแบบ ส่วนในเรื่องของการสลับแถวข้อมูลนั้น พบว่าวิธีการจัดรูปแบบข้อมูลในแต่ละแบบที่นำมาทดลอง หากไม่มีการสลับแถวข้อมูลใช้เวลาในการทำงานน้อยกว่าวิธีที่มีการสลับแถวข้อมูล จึงสรุปได้ว่า การสลับแถวข้อมูลของวิธีการกำจัดของเกาส์มีผลต่อประสิทธิภาพในการทำงาน

5.2 ข้อเสนอแนะ

.วิธีการจัดรูปแบบข้อมูลแบบใหม่นี้ หากพิจารณาในเรื่องของความคุ้มค่าในเรื่องการการใช้งานทรัพยากร จะยังเป็นวิธีที่มีการใช้งานทรัพยากรได้ไม่คุ้มค่ามากนักเนื่องจากเมื่อทำงานเสร็จในขั้นตอนหนึ่ง ๆ แล้วหน่วยประมวลผลบางตัวที่ทำงานเสร็จจะว่างงานลง ในอนาคตจึงควรปรับปรุงตรงจุดนี้ เพื่อให้สามารถใช้งานทรัพยากรได้คุ้มค่ามากขึ้น และเนื่องจากระยะเวลาในการทำวิจัยมีไม่มากนัก จึงมีการเปรียบเทียบประสิทธิภาพการทำงานของวิธีการใหม่กับวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกและวิธีการจัดรูปแบบข้อมูลแบบคอลัมน์บล็อกไฮคลิกเท่านั้น เพราะการเปรียบเทียบประสิทธิภาพนั้นต้องคำนึงถึงสถานะแวดล้อมเป็นสำคัญ การเปรียบเทียบแต่ละวิธีการจึงจำเป็นต้องต้องนำมาเขียนเป็นโปรแกรมที่มีขั้นตอนวิธีเหมือนกันและทำงานภายใต้สถานะแวดล้อมเดียวกัน ซึ่งต้องใช้เวลาในการพัฒนา ดังนั้น ในอนาคตหากต้องการให้เห็นผลที่ชัดเจนยิ่งขึ้นควรนำเอารูปแบบใหม่นี้ไปเปรียบเทียบกับวิธีการอื่น ๆ ที่ได้นำเสนอไปในบทที่ 2 ด้วย

เอกสารอ้างอิง

- [1] T. Sterling, D. Becker, D. Savarese, et al. "BEOWULF: A Parallel Workstation for Scientific Computation," Proceedings of the 1995 International Conference on Parallel Processing (ICPP). August, 1995.
- [2] Ananth G. Introduction to Parallel Computing. The Benjamin/Cummings Publishing Company, USA, 1994.
- [3] Eric W. Weisstein. "Gaussian Elimination," Matrix Operations.
<http://mathworld.wolfram.com/GaussianElimination.html>. กันยายน, 2547.
- [4] "ปรมจารย์ทางคณิตศาสตร์ ฟรีดริค เกาส์ (Carl Friedrich Gauss)," การศึกษาคณิตศาสตร์.
<http://www.tummath.com/topics2/GAUSS/Main.htm>. สิงหาคม, 2547.
- [5] I. Foster. "Case Study: Gaussian Elimination," Designing and Building Parallel Programs.
<http://www-unix.mcs.anl.gov/dbpp/text/node90.html>. กันยายน, 2547.
- [6] Seymour Lipschutz. "Solved Problems in Linear Algebra International Edition," Library of Congress Cataloging-in-Publication Data. Mcgran-Hill book company. 1989.
- [7] M. Wasilewski. Parallel Gaussian Elimination. University of Waterloo. April, 2004.
- [8] "การสลับแถวเพื่อลดความคลาดเคลื่อน," การหาผลเฉลยของระบบสมการเชิงเส้น.
http://www.rit.ac.th/homepage-sc/charud/wach/c-language/C4Physics_Web/Unit3/unit3_0.htm. ตุลาคม, 2547.
- [9] R. Brent, A. Cleary, M. Dow, M. Hegland, J. Jenkinson, Z. Leyk, M. Nakanishi, M. Os-borne, P. Price, S. Roberts, and D. Singleton. Implementation and performance of scalable sciatic library subroutines on Fujitsu's VPP500 parallel-vector supercomputer. In Proceedings of the Scalable High-Performance Computing Conference, IEEE Computer Society Press, Los Alamitos, CA. 1994.
- [10] J. Du Croz, P. Mayes, and G. Radicati. Factorization of band matrices using level 3 BLAS, Tech. Report CS-90-109, University of Tennessee, Knoxville, TN. July, 1990.

- [11] A. Gupta, F. G. Gustavson, M. Joshi, and S. Toledo. The design, implementation and evaluation of a symmetric banded linear solver for distributed-memory parallel computers, ACM Trans. Math, Software. 1998.
- [12] Bruno Lang. Parallel Direct Methods for Dense Linear Systems. Institute for Scientific Computing Aachen University of Technology. May 03, 2001.
- [13] J. W. Demmel, M. T. Heath, H. A. van der Vorst. Parallel numerical linear algebra. Computer Science Division, U.C. Berkeley, (LAPACK Working Note #60). August 18, 1996.
- [14] James W. Demmel, Michael T. Heath and Henk A. van der Vorst. Parallel numerical linear algebra. University of California at Berkeley. 1993.
- [15] J. Kuntraruk. Data Layout Issues For Efficient Parallel Gaussian Elimination. Lehigh University. December, 1999.
- [16] J.W. Demmel. "Dense Linear Algebra," Applications of Parallel Computers. http://www.cs.berkeley.edu/~demmel/cs267_221001.ppt. กันยายน, 2547.
- [17] H.D. Simon. "Large Dense Linear Systems-Distributed Implementations," Applications of Parallel Processors. <http://www.cs.berkeley.edu/cs267>. ตุลาคม, 2547.
- [18] X. S. Li, J. W. Demmel. Making Sparse Gaussian Elimination Scalable by Static Pivoting. SC98, Orlando, Florida. November 7-13, 1998.
- [19] S. Foreword, Janusz S. Kowalik. "PVM (Parallel Virtual Machine)," A users' guide and tutorial for networked parallel computing. <http://www.netlib.org/pvm3/book/node1.html>. ตุลาคม, 2547.
- [20] "MPI: A Message-Passing Interface Standard," Message Passing Interface Forum. Version 1.1, University of Tennessee, Knoxville, Tennessee. June, 1995.
- [21] William Gropp, Ewing Lusk, and Anthony Skjellum. "Using MPI," MIT Press, 1994. For more information on MPI. <http://www.msc.nsl.gov/mpi>. 1994.
- [22] Gregory D. Burns, Raja B. Daoud, and J. R. Vaigl. LAM: An open cluster environment for MPI. In Proceedings of the Supercomputer Sysposium' 94, page 379-386, Toronto, Canada. June, 1994.
- [23] A. Karp and H. Flatt. Measuring Parallel Processor Performance. Communications of the ACM, 1990.

ภาคผนวก ก
การหาผลเฉลี่ยของระบบสมการเชิงเส้นด้วยวิธีกำจัดของเกาส์

ภาคผนวก ก

การหาผลเฉลยของระบบสมการเชิงเส้นด้วยวิธีกำจัดของเกาส์

วิธีการกำจัดของเกาส์เป็นวิธีที่ใช้การหาผลเฉลยของระบบสมการเชิงเส้น โดยการจัดรูปสมการใหม่เพื่อกำจัดตัวไม่ทราบค่าในสมการเชิงเส้น ซึ่งเป็นการทำให้ค่าสัมประสิทธิ์ของตัวแปรที่อยู่ด้านใต้ของแนวเส้นทแยงมุมเป็นศูนย์โดยใช้วิธีการลดทอน เพื่อให้เข้าใจในขั้นตอนการทำงานของวิธีการกำจัดของเกาส์ที่นำมาใช้ในการแก้ปัญหาสมการเชิงเส้น จะมีตัวอย่างที่จะแสดงขั้นตอนต่าง ๆ ของการจัดตัวแปรที่ไม่ทราบค่าของสมการเชิงเส้น โดยใช้สมการ 4 สมการ 4 ตัวแปร จะอยู่ในรูปของ $Ax=b$ ดังนี้

$$5x + 2x + 1x + 1x = 9$$

$$1x + 4x + 3x + 2x = 9$$

$$3x + 5x + 12x + 4x = 17$$

$$2x + 3x + -2x + 3x = 6$$

ภาพที่ ก.1 ตัวอย่างสมการเชิงเส้น 4 สมการ

เมื่อนำมาเขียนให้อยู่ในรูปย่อจะได้ดังนี้

$$\begin{bmatrix} 5 & 2 & 1 & 1 \\ 1 & -4 & 3 & -2 \\ 3 & -5 & 12 & -4 \\ 2 & 3 & -2 & 3 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} = \begin{bmatrix} 9 \\ 9 \\ 17 \\ 6 \end{bmatrix}$$

ภาพที่ ก.2 จัดรูปแบบสมการให้อยู่ในรูปย่อของเมตริกซ์ A ขนาด 4×4 เมตริกซ์ B ขนาด 4×1

จากสมการข้างต้น เมื่อเขียนให้อยู่ในรูปแบบของเมตริกซ์แต่งเต็ม $[A | b]$ จะได้ดังนี้

$$\begin{array}{cccccc} 5 & 2 & 1 & 1 & 9 \\ 1 & -4 & 3 & -2 & 9 \\ 3 & -5 & 12 & -4 & 17 \\ 2 & 3 & -2 & 3 & 6 \end{array}$$

ภาพที่ ก.3 จัดรูปแบบสมการให้อยู่ในรูปเมตริกซ์แต่งเต็ม $[A | b]$

หลังจากจัดรูปแบบของสมการให้อยู่ในรูปแบบของเมทริกซ์แบบแคงเดิมแล้ว จะเริ่มต้นการกำจัดตามวิธีการกำจัดของเกาส์ โดยอันดับแรกจะทำการเปรียบเทียบค่าว่าจำเป็นจะต้องสลับแถวหรือไม่ โดยใช้ค่าสัมบูรณ์ของแต่ละตัวในคอลัมน์แรก

5.00	2.00	1.00	1.00	9.00
1.00	-4.00	3.00	-2.00	9.00
3.00	-5.00	12.00	-4.00	17.00
2.00	3.00	-2.00	3.00	6.00

ภาพที่ ก.4 ขั้นตอนการตรวจสอบคอลัมน์แรกว่าจำเป็นจะต้องสลับแถวหรือไม่

สำหรับสมการข้างต้นปรากฏว่าค่าสัมบูรณ์ที่มีค่ามากที่สุด ในคอลัมน์แรกอยู่ที่แถวแรก คือ 5 จึงไม่มีการสลับแถวของสมการ สามารถทำขั้นตอนต่อไปได้เลยคือการกำจัดค่า

5.00	2.00	1.00	1.00	9.00
1.00	-4.00	3.00	-2.00	9.00
3.00	-5.00	12.00	-4.00	17.00
2.00	3.00	-2.00	3.00	6.00

ภาพที่ ก.5 แถวที่ 2, 3 และ 4 คือแถวที่จะต้องถูกกำจัดค่าสัมประสิทธิ์

การกำจัดค่าสัมประสิทธิ์ของสมการแต่ละตัวจะกำจัด โดยใช้ค่าที่ได้จากขั้นตอนวิธีการ $A[j][k] = A[j][k] - (A[j][i] / A[i][i] * A[i][k])$ ผลที่ได้หลังจากการกำจัดค่าเป็นดังนี้

5.00	2.00	1.00	1.00	9.00
0.00	-4.40	2.80	-2.20	7.20
0.00	-6.20	11.40	-4.60	11.60
0.00	2.20	-2.40	2.60	2.40

ภาพที่ ก.6 ผลที่ได้จากการกำจัดค่าในรอบแรกค่าสัมประสิทธิ์ของสมการแถวที่ 2, 3 และ 4 ในคอลัมน์แรกมีค่าเป็น 0

หลังจากเสร็จสิ้นการทำงานในรอบที่ 1 แล้ว ขั้นตอนของการกำจัดสัมประสิทธิ์ของตัวแปรในรอบถัด ๆ ไปก็จะทำเช่นเดียวกันกับรอบแรก แต่ขอบเขตในการคำนวณจะลดลงตามจำนวนรอบของการทำงาน ดังแสดงในภาพที่ ก.7

5.00	2.00	1.00	1.00	9.00
0.00	-4.40	2.80	-2.20	7.20
0.00	-6.20	11.40	-4.60	11.60
0.00	2.20	-2.40	2.60	2.40

ภาพที่ ก.7 ข้อมูลที่จะต้องทำการกำจัดค่าในรอบที่ 2

เริ่มต้นรอบที่ 2 ด้วยการเปรียบเทียบหาค่าสมบูรณ์สูงสุดในคอลัมน์ที่ 2 ดังภาพที่ ก.8

5.00	2.00	1.00	1.00	9.00
0.00	-4.40	2.80	-2.20	7.20
0.00	-6.20	11.40	-4.60	11.60
0.00	2.20	-2.40	2.60	2.40

ภาพที่ ก.8 เปรียบเทียบเพื่อหาค่าสมบูรณ์สูงสุดในคอลัมน์ที่ 2 ค่าสมบูรณ์สูงสุดคือ 6.20

สำหรับในกรณีนี้ปรากฏว่าแถวที่ 3 มีค่าสมบูรณ์สูงสุดคือ 6.20 ฉะนั้นจะต้องมีการสลับแถวของสมการเกิดขึ้น โดยจะทำการสลับแถวระหว่างแถวที่ 2 กับแถวที่ 3 เมื่อทำการสลับแถวเสร็จแล้วจะได้ผลดังแสดงในรูปภาพที่ ก.9

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	-4.40	2.80	-2.20	7.20
0.00	2.20	-2.40	2.60	2.40

ภาพที่ ก.9 สลับแถวของสมการระหว่างแถวที่ 2 กับแถวที่ 3

การกำจัดสัมประสิทธิ์ของตัวแปรในแถวที่ 2 และ 3 ดังแสดงในภาพที่ ก.10

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	-4.40	2.80	-2.20	7.20
0.00	2.20	-2.40	2.60	2.40

ภาพที่ ก.10 แถวที่จะต้องทำการกำจัดค่าสัมประสิทธิ์ คือแถวที่ 3 และ 4

ผลที่ได้จากการกำจัดค่าสัมประสิทธิ์ แสดงดังนี้

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	0.00	-5.29	1.65	-1.03
0.00	0.00	1.65	0.97	6.52

ภาพที่ ก.11 ผลที่ได้จากการกำจัดค่าสัมประสิทธิ์ของตัวแปรในรอบที่ 2

ในรอบสุดท้ายของขั้นตอนที่ทำตามวิธีการที่ผ่านมาในรอบที่ 1 และรอบที่ 2

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	0.00	-5.29	1.65	-1.03
0.00	0.00	1.65	0.97	6.52

ภาพที่ ก.12 ข้อมูลที่จะต้องกำจัดค่าในรอบสุดท้าย

โดยจะทำการเปรียบเทียบเพื่อหาค่าสมบูรณ์สูงสุดในคอลัมน์ที่ 3 ระหว่างแถวที่ 2 กับ 3

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	0.00	-5.29	1.65	-1.03
0.00	0.00	1.65	0.97	6.52

ภาพที่ ก.13 เปรียบเทียบเพื่อหาค่าสมบูรณ์สูงสุด

หลังจากได้ทำการเปรียบเทียบค่าสมบูรณ์สูงสุดอยู่ที่แถวที่ 3 คือ 5.29 จึงไม่ต้องการสลับแถว สามารถเริ่มทำการกำจัดสัมประสิทธิ์ของตัวแปรได้ทันที

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	0.00	-5.29	1.65	-1.03
0.00	0.00	0.00	1.30	6.20

ภาพที่ ก.14 ผลที่ได้จากการกำจัดรอบสุดท้าย

หลังจากสิ้นสุดการกำจัดในรอบสุดท้าย ค่าต่าง ๆ ในสามเหลี่ยมด้านล่างจะเป็น 0 ดังแสดง
ในภาพที่ ก.15

5.00	2.00	1.00	1.00	9.00
0.00	-6.20	11.40	-4.60	11.60
0.00	0.00	-5.29	1.65	-1.03
0.00	0.00	0.00	1.30	6.20

ภาพที่ ก.15 ผลที่ได้เมื่อสิ้นสุดการทำงาน

หลังจากสิ้นสุดขั้นตอนการกำจัดค่าแล้วจะทำการแทนค่าย้อนกลับโดยใช้สมการที่ 1 จะได้ผลของตัวแปรต่าง ๆ ดังนี้

$$\begin{aligned}
 x^1 &= 1.92 \\
 x^2 &= -3.28 \\
 x^3 &= 1.15 \\
 x^4 &= 4.77
 \end{aligned}$$

ประวัติผู้วิจัย

ชื่อ

นายสมหมาย ศรีบัวบาล

ประวัติการศึกษา

สถาบันราชภัฏอุบลราชธานี, พ.ศ. 2540-2544

วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยอุบลราชธานี, พ.ศ. 2546-2548

วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ประวัติการทำงาน

โปรแกรมเมอร์ บริษัท ราชบุตรเทเลคอม จำกัด

โปรแกรมเมอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี

ที่ปรึกษาด้านการพัฒนาซอฟต์แวร์ บริษัท โปรคลิก จำกัด

ทุนสนับสนุนการทำวิทยานิพนธ์จากสำนักงานคณะกรรมการ

อุดมศึกษาแห่งชาติ (สกอ.)

ทุนพัฒนาอาจารย์สาขาขาดแคลน สกอ. (ปริญญาโท-เอก)

สาขาวิชาเทคโนโลยีสารสนเทศ คณะบริหารศาสตร์

มหาวิทยาลัยอุบลราชธานี ปี 2547