

การประยุกต์ใช้การโปรแกรมคำนวณแบบขนานลูกผสมกับการจำลองสึนามิ

แบบหลายระดับความละเอียด



นายธารนิจ เศรษฐพานิชผล

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

APPLYING HYBRID PARALLEL PROGRAMMING TO MULTI-SCALE TSUNAMI SIMULATION

Mr. Tanniti Setapanichpol



จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การประยุกต์ใช้การโปรแกรมคำนวณแบบขนานลูกผสมกับการ  
จำลองสึนามิแบบหลายระดับความละเอียด

โดย

นายธารนิจ เศรษฐพานิชผล

สาขาวิชา

วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมืองสิน

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.บุญสม เลิศศิริวงค์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ณัฐฉิ หนูไพโรจน์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมืองสิน)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.ภูชงค์ อุทโยภาศ)

ธารนิธิ เศรษฐพานิชผล : การประยุกต์ใช้การโปรแกรมคำนวณแบบขนานลูกผสมกับการจำลองสึนามิแบบหลายระดับความละเอียด. (APPLYING HYBRID PARALLEL PROGRAMMING TO MULTI-SCALE TSUNAMI SIMULATION) อ.ที่ปรึกษาวิทยานิพนธ์  
 หลัก: ผศ. ดร.วีระ เหมือนสิน, 109 หน้า.

ในปัจจุบันคอมพิวเตอร์แบบขนานมักถูกสร้างขึ้นมาจาก 3 เทคโนโลยีอันได้แก่ คลัสเตอร์คอมพิวเตอร์ (Cluster computer), มัลติโพรเซสเซอร์หรือมัลติคอร์ (Multiprocessors/Multicores) และหน่วยประมวลผลกราฟิกส์สำหรับงานทั่วไป (General Purpose Graphics Processing Unit/GPGPU) ซึ่งมีความแตกต่างกันหลายประการ เช่น ลำดับชั้นของการทำงานแบบขนาน (level of parallelism), สถาปัตยกรรม, อัลกอริทึม, รูปแบบการเขียนโปรแกรม และความสามารถในการปรับขนาด (scalability) เป็นต้น ดังนั้นเพื่อที่จะใช้ประโยชน์จากการทำงานแบบขนานในหลายระดับและเพิ่มความสามารถในการปรับขนาดคอมพิวเตอร์แบบขนานหนึ่งเครื่องอาจสร้างมาจากการรวมกันของทั้ง 3 เทคโนโลยีข้างต้น เกิดเป็นคอมพิวเตอร์ขนานแบบลูกผสม คือ เป็นคลัสเตอร์คอมพิวเตอร์ที่มีหน่วยประมวลผลแบบมัลติคอร์และหน่วยประมวลผลกราฟิกส์ ซึ่งจะนำเอาข้อดีของแต่ละเทคนิคมาใช้เพื่อเพิ่มประสิทธิภาพในการทำงานของโปรแกรมโดยรวม งานวิจัยนี้จึงต้องการหาวิธีจัดการงานของโปรแกรมคำนวณแบบขนาน สำหรับปัญหาที่มีหลายระดับความละเอียด ที่ทำงานบนคลัสเตอร์คอมพิวเตอร์ที่มีหน่วยประมวลผลกราฟิกส์ ด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม โดยแบ่งงานและกระจายงานไปยังหน่วยประมวลผลที่ไม่สมมาตรอย่างเหมาะสม และสามารถปรับการจัดการงานให้เหมาะสมกับความสามารถของระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้งาน โดยใช้การจำลองสึนามิเป็นกรณีศึกษา เพื่อที่จะนำวิธีการจัดการงานที่ได้ไปประยุกต์ใช้กับโปรแกรมคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียดอื่นๆ

จุฬาลงกรณ์มหาวิทยาลัย  
 CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2555.....

# # 5470228021 : MAJOR COMPUTER ENGINEERING

KEYWORDS: PARALLEL COMPUTING / HYBRID PARALLEL PROGRAMMING / GPU / MPI / GRAPHIC PROCESSING UNIT / MESSAGE PASSING INTERFACE / ASYMMETRIC MULTICORE PROCESSORS

TANNITI SETAPANICHPOL: APPLYING HYBRID PARALLEL PROGRAMMING TO MULTI-SCALE TSUNAMI SIMULATION. ADVISOR : ASST.PROF. VEERA MUANGSIN, Ph.D., 109 pp.

Currently, most parallel computers are built from three technologies, namely cluster computer, multiprocessor/multicore, and general purpose graphics processing unit (GPGPU). In order to exploit many levels of parallelism and improve scalability, many parallel computers are hybrid systems built from combinations of the three. Mixing their programming interfaces in a non-trivial program is quite challenging. Moreover, the introduction of GPUs brings new problems in load balancing because now the processors (CPUs/GPUs) are asymmetric. This paper presents the development of a hybrid parallel program using MPI and CUDA for a cluster of multicore processors and GPUs. The target application is a tsunami simulation program called TUNAMI. We especially focus on partitioning techniques for load balancing. The work can be divided into two parts. The first part was to develop a hybrid version of the program based on an MPI version. The second part was to develop an analysis tool to support data partitioning onto CPUs and GPUs.

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

Department: Computer Engineering... Student's Signature .....

Field of Study: Computer Engineering... Advisor's Signature .....

Academic Year: 2012.....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดีด้วยความช่วยเหลือของบุคคลหลายท่าน ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมือนสิน อาจารย์ที่ปรึกษาวิทยานิพนธ์หลักซึ่งได้ช่วยสละเวลาให้คำชี้แนะต่างๆตลอดจนให้ความช่วยเหลือในการแก้ไขปัญหาต่างๆ ตลอดระยะเวลาในการทำวิทยานิพนธ์ฉบับนี้ อีกทั้งยังได้ให้แนวคิดที่ดีสำหรับการใช้ชีวิต

ลำดับถัดมาขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูโพโรจน์ ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา กรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.ภูษงค์ อุทัยภาส กรรมการภายนอกมหาวิทยาลัย ที่ได้กรุณาแนะนำและชี้แนะแนวทางที่เป็นประโยชน์ต่อการทำวิทยานิพนธ์ในครั้งนี้

ขอกราบขอบพระคุณ ดร.ศุภกิจ พฤกษ์อรุณ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค) ที่ได้กรุณาให้คำแนะนำอย่างละเอียดเกี่ยวกับการดำเนินการติดตั้งระบบคลัสเตอร์สำหรับใช้ในการทดลองของวิทยานิพนธ์ในครั้งนี้ นายอลงกต บุรุษอาชาไนย ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้คำแนะนำสำหรับการทำวิทยานิพนธ์ แนวคิดในการทำงาน ทรรศนะคติ อีกทั้งยังให้คำปรึกษาที่ดีในเวลาที่เกิดปัญหาหรือข้อสงสัย รวมถึงขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ ร่วมห้องปฏิบัติการและนิสิตปริญญาโททุกคน ที่ช่วยเหลือในการดำเนินชีวิตในมหาวิทยาลัยแห่งนี้เป็นอย่างดี โดยเฉพาะนายพงพัฒน์ เป้าเพชร ที่ได้คอยให้ทั้งความช่วยเหลือและรับฟังปัญหาต่างๆ ที่เกิดขึ้นไม่ว่าจะเป็นเรื่องการศึกษาหรือการดำเนินชีวิต รวมถึงการแลกเปลี่ยนทรรศนะคติต่างๆ ทำให้มีความสุขและสนุกสนานกับการศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์ตลอด 2 ปีที่ผ่านมา

สุดท้ายนี้ขอกราบขอบพระคุณ นายชำนาญ เศรษฐพานิชผล และนางนันทมน เศรษฐพานิชผล บิดามารดา ที่ได้ให้กำเนิดและเลี้ยงดูเป็นอย่างดี รวมทั้งสมาชิกในครอบครัวที่คอยกระตุ้นและให้กำลังใจเสมอมา และขอกราบขอบพระคุณคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาตลอดช่วงชีวิตที่ผ่านมา ทำให้มีความรู้ติดตัวจนถึงทุกวันนี้

## สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฎ
สารบัญรูป.....	ฏ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	4
1.3 ขอบเขตของการวิจัย.....	4
1.4 ขั้นตอนและวิธีดำเนินการวิจัย.....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	5
1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์.....	5
1.7 ผลงานที่ได้รับการตีพิมพ์.....	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 การคำนวณแบบขนาน (Parallel Computing).....	6
2.1.1.1 Message Passing Model.....	7
2.1.1.2 MPI (Message Passing Interface).....	7
2.1.1.3 CUDA (Compute Unified Device Architecture).....	8
2.1.1.4 Hybrid Parallel Programming.....	11
2.1.1.5 Data-Parallel model.....	12
2.1.1.6 Asymmetric Multi Processors.....	13

2.1.1.7 ปัญหาที่มีหลายระดับความละเอียด (Multi-scale Problem).....	13
2.1.2 การปรับแต่งประสิทธิภาพของโปรแกรมคำนวณแบบขนาน (Performance Tuning of Parallel Program) .....	13
2.1.2.1 การแบ่งงานและกระจายงาน (Partitioning and Load Balancing).....	13
2.1.2.2 Zero-Copy Memory .....	14
2.1.3 การวัดประสิทธิภาพของโปรแกรมคำนวณแบบขนาน (Performance Measuring of Parallel Program) .....	14
2.1.3.1 ความเร็วที่เพิ่มขึ้น (Speedup) .....	14
2.1.3.2 ประสิทธิภาพของการทำงาน (Efficiency).....	15
2.1.4 การเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนาน (Program Profiling) .....	16
2.1.4.1 IPM Version 0.9.80 (Integrated Performance Monitoring) and Ploticus Version 2.4.1 .....	16
2.2 งานวิจัยที่เกี่ยวข้อง.....	16
2.2.1 การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่มีหลายระดับความละเอียด : กรณีศึกษาการจำลองสึนามิ (Parallel program development with Adaptive Performance Tuning for Multi-Scale problem : A case study of tsunami simulation).....	16
2.2.2 ระบบเตือนภัยสึนามิด้วยการจำลองแบบทันกาลบนหน่วยประมวลผลกราฟิกส์ (Tsunami Warning System with Real-Time Simulation on Graphics Processing Unit) .....	21
2.2.3 Parallelizing TUNAMI-N1 using GPGPU .....	24
2.2.4 Asynchronous Communication Schemes for Finite Difference Methods on Multiple GPUs.....	26
บทที่ 3 กรณีศึกษา .....	28



3.1	กรณีศึกษาที่ใช้ในงานวิจัย .....	28
3.1.1	โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับ (TUNAMI Program).....	28
3.1.2	โปรแกรมแบบขนานเพื่อจำลองการเกิดสึนามิ (Parallel Tsunami Simulation Program) .....	33
บทที่ 4	แนวคิดการทำวิจัย .....	37
4.1	แนวคิดหลัก .....	37
4.1.1	แนวคิดในการออกแบบวิธีการจัดการงานแบบขนาน .....	37
4.1.2	แนวคิดในการทดสอบ .....	38
4.1.3	แนวคิดในการประเมินผล .....	39
4.2	การพัฒนาโปรแกรมคำนวณแบบขนานด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม .....	39
4.2.1	การพัฒนาส่วนโค้ดหลัก .....	39
4.2.2	การคอมไพล์โปรแกรมแบบขนานลูกผสม .....	41
4.2.3	การเรียกใช้งานโปรแกรมแบบขนานลูกผสม .....	42
4.3	การวิเคราะห์ประสิทธิภาพของโปรแกรมที่คำนวณแบบขนาน .....	42
4.3.1	การวิเคราะห์ประสิทธิภาพเชิงเวลาของแต่ละส่วนการทำงานของโปรแกรม .....	42
4.3.1.1	การวิเคราะห์โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานและขนานลูกผสม.....	43
4.3.2	การวิเคราะห์ประสิทธิภาพเชิงเวลาที่ขึ้นกับขนาดของปัญหา .....	46
4.3.3	การวิเคราะห์การติดต่อสื่อสารระหว่างหน่วยประมวลผล.....	47
4.3.3.1	การแลกเปลี่ยนข้อมูลภายนอกของแต่ละระดับความละเอียด (Inter Communication) .....	47
4.3.3.2	การแลกเปลี่ยนข้อมูลภายในของแต่ละระดับความละเอียด (Intra Communication) .....	49

4.3.3.3 การแลกเปลี่ยนข้อมูลระหว่างระดับความละเอียด เมื่อมีการแบ่งงานภายในระดับความละเอียด.....	51
4.3.3.4 การรรับข้อมูลของการติดต่อสื่อสารระหว่างหน่วยประมวลผล.....	52
4.4 การแบ่งงานให้เหมาะสมบนระบบคลัสเตอร์คอมพิวเตอร์แบบไฮบริด.....	53
4.4.1 การแบ่งงานให้เหมาะสมโดยปรับสัดส่วนของงานระหว่างหน่วยประมวลผล.....	53
4.4.2 การจัดสรรงานให้เหมาะสมตามข้อกำหนดในเรื่องของเวลาและหน่วยประมวลผลที่ใช้งาน.....	55
บทที่ 5 การทดลองและวิเคราะห์ผลการวิจัย.....	57
5.1 การทดลอง.....	57
5.1.1 รูปแบบการทดลอง.....	57
5.1.1.1 ข้อมูลที่ใช้ในการทดลอง.....	57
5.1.1.2 เครื่องมือที่ใช้ในการทดลอง.....	58
5.1.2 ผลการทดลอง.....	60
5.1.2.1 การพัฒนาโปรแกรมคำนวณแบบขนานด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม.....	60
5.1.2.2 ผลการทำงานของโปรแกรมแบบขนานลูกผสม.....	66
5.2 การวิเคราะห์ผล.....	68
5.2.1 การวิเคราะห์ผลโปรแกรมจำลองสีนามิที่คำนวณแบบขนานลูกผสม.....	68
5.2.1.1 ส่วนการทำงานที่เวลาขึ้นกับขนาดปัญหาทั้งหมดในแต่ละระดับความละเอียด.....	69
5.2.1.2 ส่วนการทำงานที่เวลาขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล.....	71
5.2.1.3 ส่วนการทำงานที่เวลาขึ้นกับขนาดของปัญหาเฉพาะพื้นที่น้ำในแต่ละหน่วยประมวลผล.....	75
5.2.1.4 ส่วนการทำงานที่ขึ้นกับปริมาณข้อมูลของการติดต่อสื่อสารระหว่างหน่วยประมวลผลในแต่ละครั้ง.....	77

5.3 สรุปผลการวิเคราะห์ .....	79
5.3.1 สรุปผลการวิเคราะห์การทำงานของโปรแกรมจำลองสึนามิที่คำนวณแบบ ขนานลูกผสม .....	79
5.3.2 สรุปผลการวิเคราะห์การสื่อสารของโปรแกรมจำลองสึนามิแบบขนาน ลูกผสม .....	83
5.3.3 ลำดับการทำงานของโปรแกรมแบบขนานลูกผสม .....	89
บทที่ 6 การประเมินผลการทดลอง.....	91
6.1 การประเมินผล.....	91
6.1.1 ข้อมูลที่ใช้ในการประเมินผล.....	91
6.1.2 เครื่องมือที่ใช้ในการประเมินผล.....	91
6.1.3 รูปแบบของการประเมินผล .....	91
6.2 ผลการประเมิน.....	92
6.2.1 ประสิทธิภาพจากการแบ่งงาน.....	92
6.2.2 ความแม่นยำของผลลัพธ์.....	95
6.2.3 เวลาที่ใช้ในการคำนวณจริงกับเวลาที่คำนวณจากสูตรคำนวณเวลาที่ได้.....	96
6.3 สรุปผลการประเมิน.....	98
บทที่ 7 สรุปผลการวิจัย.....	99
7.1 สรุปผลการวิจัย .....	99
7.2 แนวทางในการวิจัยต่อ.....	100
รายการอ้างอิง .....	102
ภาคผนวก.....	104
ภาคผนวก ก .....	105
ภาคผนวก ข .....	108
ประวัติผู้เขียนวิทยานิพนธ์ .....	109

สารบัญตาราง

หน้า

ตารางที่ 2-1	พิกัดและขนาดของพื้นที่ในแต่ละระดับความละเอียดที่ใช้ในการทดสอบสำหรับโปรแกรมจำลองสึนามิที่คำนวณแบบขนาน .....	18
ตารางที่ 2-2	ตารางเปรียบเทียบอัลกอริทึมในการทำงานระหว่างการคำนวณพื้นที่ บนหน่วยประมวลผลกราฟิกส์.....	27
ตารางที่ 3-1	พิกัดของบริเวณที่นำมาคำนวณด้วยโปรแกรมจำลองสึนามิ.....	31
ตารางที่ 4-1	เวลาที่ใช้ของแต่ละส่วนการทำงานและเวลาที่ใช้ในการทำงานทั้งหมดของโปรแกรม TUNAMI .....	40
ตารางที่ 4-2	ขนาดของปัญหาในแต่ละระดับความละเอียดทั้งหมด.....	46
ตารางที่ 5-1	พิกัดและขนาดของแต่ละพื้นที่ในแต่ละระดับความละเอียดที่ใช้ในการทดลองทั้งโปรแกรมที่คำนวณแบบขนานและแบบขนานลูกผสม.....	57
ตารางที่ 5-2	ชื่องานและจำนวนของหน่วยประมวลผลที่ใช้งานของแต่ละชุดข้อมูลที่ใช้ในการทดลองโปรแกรมจำลองสึนามิทั้งที่คำนวณแบบขนานและแบบขนานลูกผสม .....	58
ตารางที่ 5-3	ผลการทดสอบค่าสัดส่วนประสิทธิภาพระหว่างหน่วยประมวลผลกลางเทียบกับหน่วยประมวลผลกราฟิกส์เมื่อทดลองด้วยส่วนการทำงานแบบต่างๆ บนคลัสเตอร์คอมพิวเตอร์.....	60
ตารางที่ 5-4	ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบเพกาซัสคลัสเตอร์.....	62
ตารางที่ 5-5	ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบสเปซคลัสเตอร์ .....	63
ตารางที่ 5-6	ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบเพกาซัสคลัสเตอร์.....	64
ตารางที่ 5-7	ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบสเปซคลัสเตอร์.....	65
ตารางที่ 5-8	ค่าตัวแปรในสมการเชิงเส้นของส่วนการทำงานของโปรแกรมจำลองสึนามิแบบขนานลูกผสม .....	79
ตารางที่ 5-9	ผลการประมาณเวลาที่ใช้ในการทำงานของแต่ละระดับความละเอียดของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม.....	80

ตารางที่ 5-10 แสดงผลการเปรียบเทียบเวลาที่ใช้ในการคำนวณปัญหาส่วนของพื้หน้า .....	81
ตารางที่ 5-11 ค่าตัวแปรที่ใช้สำหรับการหาปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสารภายใน ระดับความละเอียดของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม .....	83
ตารางที่ 5-12 ค่าตัวแปรที่ใช้สำหรับการหาปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่าง ระดับความละเอียดของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม .....	83
ตารางที่ 5-13 ปริมาณข้อมูล จำนวนครั้งในการรับส่งข้อมูล และเวลาที่ใช้ในการสื่อสาร ภายในระดับความละเอียดของโปรแกรม .....	84
ตารางที่ 5-14 ปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่างระดับความละเอียด .....	86
ตารางที่ 5-15 เวลาที่ใช้ในการติดต่อสื่อสารภายนอกระดับความละเอียด .....	87
ตารางที่ 5-16 ลำดับของการทำงานที่มีการรอรับข้อมูลของโปรแกรมจำลองสึนามิที่คำนวณ แบบขนานลูกผสม .....	89
ตารางที่ 5-17 ลำดับของการทำงานที่มีการรอรับข้อมูลของโปรแกรมจำลองสึนามิที่คำนวณ แบบขนานลูกผสมที่คำนวณหลายโชนพร้อมกัน .....	90
ตารางที่ 6-1 การจัดเรียงข้อมูลเพื่อหาจำนวนของหน่วยประมวลผลที่ใช้งานจริง .....	92
ตารางที่ 6-2 ตารางเปรียบเทียบการแบ่งงานในรูปแบบปกติซึ่งทุกระดับความละเอียดใช้ จำนวนหน่วยประมวลผลในการคำนวณเท่ากัน (CUDA-Optimize) กับการ เฉลี่ยงานตามประสิทธิภาพของหน่วยประมวลผล (CUDA-Optimize- Partitioning) .....	92
ตารางที่ 3-6 ตารางแสดงความเร็วที่เพิ่มขึ้นจากการปรับปรุงโปรแกรมจำลองสึนามิ .....	94
ตารางที่ 6-4 การเปรียบเทียบผลลัพธ์ที่ได้จากการคำนวณเพื่อแสดงความแม่นยำของ ผลลัพธ์จากการใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ .....	95
ตารางที่ 6-5 ตารางแสดงผลของการประมาณเวลาในการทำงานของโชน 2111 .....	96
ตารางที่ 6-6 ตารางแสดงผลของการประมาณเวลาในการทำงานของโชน 2112 .....	96
ตารางที่ 6-7 ตารางแสดงผลของการประมาณเวลาในการทำงานของโชน 2121 .....	97
ตารางที่ 6-8 ตารางแสดงผลของการประมาณเวลาในการทำงานของโชน 2211 .....	97
ตารางที่ 6-9 เวลาที่ใช้ทำงานจริงของข้อมูลทดสอบโดยระบบเพกซัสคลัสเตอร์ .....	97
ตารางที่ 10-6 ตารางแสดงประสิทธิภาพและความเร็วที่เพิ่มขึ้นจากการปรับปรุงโปรแกรม จำลองสึนามิ .....	98

## สารบัญรูป

หน้า

รูปที่ 1-1	รูปแบบของปัญหาที่มีหลายระดับความละเอียด.....	3
รูปที่ 2-1	รูปแบบการคำนวณแบบลำดับ [1].....	6
รูปที่ 2-2	รูปแบบการคำนวณแบบขนาน [1].....	6
รูปที่ 2-3	ตัวอย่างของการสื่อสารระหว่างหน่วยประมวลผลด้วยการส่งข้อความ [1] .....	7
รูปที่ 2-4	เปรียบเทียบสถาปัตยกรรมระหว่าง CPU และ GPU [6].....	8
รูปที่ 2-5	ขั้นตอนการประมวลผลบน CUDA [6].....	10
รูปที่ 2-6	หลักการทำงานของ CUDA [6].....	10
รูปที่ 2-7	โครงสร้างของหน่วยประมวลผลบนหน่วยประมวลผลกราฟิกส์ [6].....	11
รูปที่ 2-8	ตัวอย่างการทำงานของโปรแกรมแบบขนานลูกผสม [7, 8].....	12
รูปที่ 2-9	รูปแบบของ Data-Parallel model [1].....	12
รูปที่ 2-10	การทดสอบเพื่อใช้ในการวิเคราะห์ข้อมูล .....	18
รูปที่ 2-11	ตัวอย่างการปรับปรุงส่วนการทำงาน ZMAX ให้ทำงานด้วย CUDA Fortran.....	22
รูปที่ 2-12	ตัวอย่างการเรียกใช้ส่วนการทำงาน ZMAX ให้ทำงานบนหน่วยประมวลผล กราฟิกส์ .....	22
รูปที่ 2-13	ลำดับการทำงานของโปรแกรมที่คำนวณบนหน่วยประมวลผลกราฟิกส์ .....	23
รูปที่ 2-14	ลำดับการทำงานของแบบจำลองสึนามิ Data-parallel ที่คำนวณบนหน่วย ประมวลผลกราฟิกส์ .....	25
รูปที่ 2-15	ลำดับการทำงานของแบบจำลองสึนามิ Hybrid ที่คำนวณบนหน่วยประมวลผล กราฟิกส์ .....	25
รูปที่ 2-16	ตัวอย่างการแบ่งพื้นที่ออกจากพื้นที่หลักเพื่อใช้งานบนหน่วยประมวลผลกราฟิกส์ .....	27
รูปที่ 3-1	ตัวอย่างรูปแบบการวิเคราะห์เชิงตัวเลขแบบ Finite Difference Method.....	29
รูปที่ 3-2	รายละเอียดของพื้นที่ที่ใช้งานในโปรแกรม TUNAMI .....	30
รูปที่ 3-3	ลำดับการทำงานของโปรแกรมคำนวณแบบลำดับในส่วนที่มีการเชื่อมต่อกันของ ข้อมูล .....	34
รูปที่ 3-4	ลำดับการทำงานของโปรแกรมด้วยการคำนวณแบบขนาน .....	35
รูปที่ 3-5	ตัวอย่างการแบ่งการคำนวณของแต่ละระดับความละเอียดออกเป็นงานย่อยๆ .....	35
รูปที่ 3-6	โครงสร้างแบบต้นไม้ของข้อมูลของโปรแกรมจำลองสึนามิแบบขนาน .....	36

รูปที่ 3-7 ตัวอย่างวิธีการ Page flipping .....	36
รูปที่ 4-1 ตัวอย่างการเรียกใช้คำสั่งเพื่อคำนวณงานบนหน่วยประมวลผลกราฟิกส์ .....	41
รูปที่ 4-2 ตัวอย่างการคอมไพล์โปรแกรมแบบขนานลูกผสม .....	42
รูปที่ 4-3 ตัวอย่างการเรียกใช้งานโปรแกรมแบบขนานลูกผสม .....	42
รูปที่ 4-4 การเก็บข้อมูลการทำงานของโปรแกรมด้วยการแทรกคำสั่ง MPI_Wtime.....	43
รูปที่ 4-5 บริเวณที่ระดับ R1 ต้องส่งค่าตัวแปร m_jnq (ซ้าย) และ ค่าตัวแปร n_jnq (ขวา) ให้ระดับ R2 .....	47
รูปที่ 4-6 บริเวณที่ระดับ R2 และ R3 ต้องส่งค่าตัวแปร dzb ให้ระดับ R3 และ R4 .....	48
รูปที่ 4-7 บริเวณที่ระดับ R2 และ R3 ต้องส่งค่าตัวแปร mb (ซ้าย) และ nb (ขวา) ให้ระดับ R3 และ R4 .....	48
รูปที่ 4-8 บริเวณที่ระดับ R3 และ R4 ต้องส่งค่าตัวแปร dzb และ ค่าตัวแปร zb ให้ระดับ R2 และ R3 .....	49
รูปที่ 4-9 บริเวณที่ส่วนย่อยในระดับ R1 ต้องส่งค่าตัวแปร ma (ซ้าย) และ ค่าตัวแปร na (ขวา) ให้ส่วนย่อยที่อยู่ติดกันในระดับ R1 .....	49
รูปที่ 4-10 บริเวณที่ส่วนย่อยในระดับ R1 ต้องส่งค่าตัวแปร zb ให้ส่วนย่อยที่อยู่ติดกันในระดับ R1 .....	50
รูปที่ 4-11 บริเวณที่ส่วนย่อยในระดับ R2, R3, R4 ต้องส่งค่าตัวแปร ma และ na ให้ส่วนย่อยที่อยู่ติดกันในระดับ R2, R3, R4.....	50
รูปที่ 4-12 บริเวณที่ส่วนย่อยในระดับ R2, R3, R4 ต้องส่งค่าตัวแปร dzb (ซ้าย) และ ค่าตัวแปร za, zb (ขวา) ให้ส่วนย่อยที่อยู่ติดกันในระดับ R2, R3, R4.....	51
รูปที่ 4-13 บริเวณที่ส่วนย่อยในระดับ R2, R3, R4 ต้องส่งค่าตัวแปร dma และ dna ให้ส่วนย่อยที่อยู่ติดกันในระดับ R2, R3, R4.....	51
รูปที่ 4-14 การแลกเปลี่ยนข้อมูลภายนอกความละเอียด เมื่อมีการแบ่งงานภายในระดับความละเอียด .....	52
รูปที่ 4-15 การรอรับข้อมูลระหว่างหน่วยประมวลผล .....	52
รูปที่ 4-16 แผนภาพของโปรแกรมแบบขนานลูกผสมสำหรับงานวิจัยนี้ .....	53
รูปที่ 5-1 ผลการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบเพกาซัสคลัสเตอร์ .....	61

รูปที่ 5-2	ผลการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบสเปคัลลิสเตอร์.....	62
รูปที่ 5-3	ผลการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบเพกาซัสคลัสเตอร์ .....	64
รูปที่ 5-4	ผลการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบสเปคัลลิสเตอร์ .....	65
รูปที่ 5-5	ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบหนึ่งโชน .....	66
รูปที่ 5-6	ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบสองโชน .....	67
รูปที่ 5-7	ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบสามโชน .....	67
รูปที่ 5-8	ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบสี่โชน.....	68
รูปที่ 5-9	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน INITIAL.....	69
รูปที่ 5-10	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน JNQ_S2C.....	70
รูปที่ 5-11	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน JNZ .....	70
รูปที่ 5-12	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน JNQ.....	70
รูปที่ 5-13	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน INTERQT ที่ใช้หน่วยประมวลผลกลางในการคำนวณ .....	71
รูปที่ 5-14	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน INTERQT ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ .....	72
รูปที่ 5-15	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน MAX.....	72
รูปที่ 5-16	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน ETACH ที่ใช้หน่วยประมวลผลกลางในการคำนวณ.....	72
รูปที่ 5-17	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน ETACH ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ.....	73
รูปที่ 5-18	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP_3 ที่ใช้หน่วยประมวลผลกลางในการคำนวณ.....	73
รูปที่ 5-19	ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP_3 ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ .....	74



รูปที่ 5-20 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน OUTB .....	74
รูปที่ 5-21 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน OUTMAX .....	75
รูปที่ 5-22 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน MAMO.....	76
รูปที่ 5-23 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP_1 ที่ใช้หน่วยประมวลผลกลางในการคำนวณ.....	76
รูปที่ 5-24 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP_1 ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ .....	76
รูปที่ 5-25 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP_2 ที่ใช้หน่วยประมวลผลกลางในการคำนวณ.....	77
รูปที่ 5-26 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP_2 ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ .....	77
รูปที่ 5-27 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของ ข้อมูลที่ใช้ในการสื่อสารของ ส่วนการทำงาน Send .....	78
รูปที่ 5-28 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของ ข้อมูลที่ใช้ในการสื่อสารของ ส่วนการทำงาน Recv.....	78
รูปที่ 6-1 แสดงเวลาที่ใช้ในการทำงานของโชน 2111 ของรูปแบบการทำงานต่างๆ .....	93
รูปที่ 6-2 รายละเอียดของเวลาที่ใช้ในการทำงานของโปรแกรมแบบขนานแบบต่างๆ.....	95

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของปัญหา

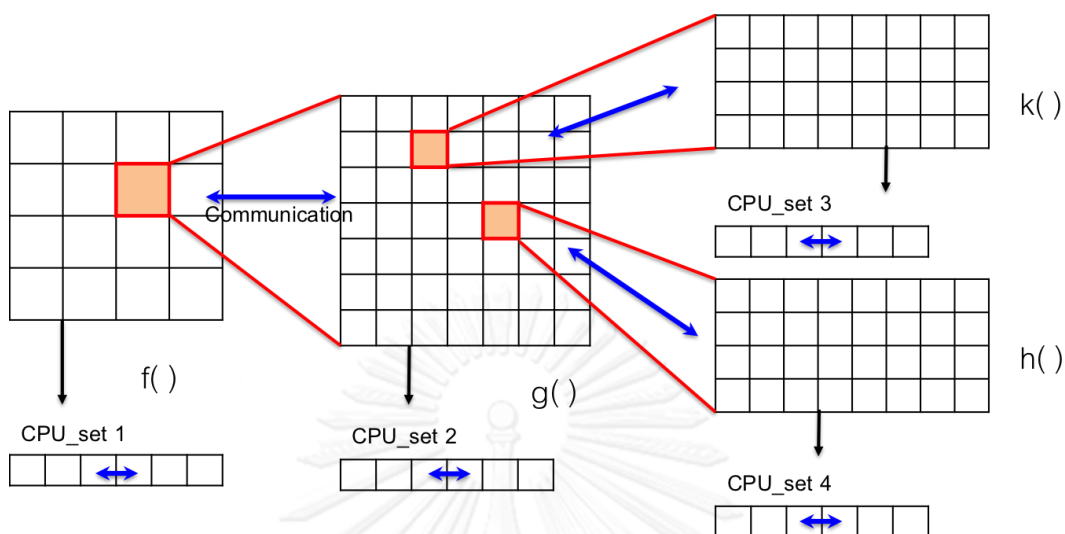
การคำนวณแบบขนาน (Parallel Computing) เป็นการใช้งานพร้อมกันของหลายทรัพยากรคำนวณเพื่อที่จะแก้ปัญหาทางการคำนวณโดยมีเป้าหมายเพื่อลดเวลาในการทำงาน [1] ในปัจจุบันคอมพิวเตอร์แบบขนานมักถูกสร้างขึ้นมาจาก เทคโนโลยีอันได้แก่ 3 คลัสเตอร์คอมพิวเตอร์ (Cluster computer), มัลติโพรเซสเซอร์หรือมัลติคอร์ (Multiprocessors/Multicores) และหน่วยประมวลผลกราฟิกส์สำหรับงานทั่วไป (General Purpose Graphics Processing Unit/GPGPU) ซึ่งมีความแตกต่างกันหลายประการ เช่น ลำดับชั้นของการทำงานแบบขนาน (level of parallelism), สถาปัตยกรรม, อัลกอริทึม, รูปแบบการเขียนโปรแกรม และความสามารถในการปรับขนาด (scalability) เป็นต้น ดังนั้นเพื่อที่จะใช้ประโยชน์จากการทำงานแบบขนานในหลายระดับและเพิ่มความสามารถในการปรับขนาดคอมพิวเตอร์แบบขนานหนึ่งเครื่องอาจสร้างมาจากการรวมกันของทั้ง 3 เทคโนโลยีข้างต้น เกิดเป็นคอมพิวเตอร์ขนานแบบลูกผสม คือ เป็นคลัสเตอร์คอมพิวเตอร์ที่มีหน่วยประมวลผลแบบมัลติคอร์และหน่วยประมวลผลกราฟิกส์ ซึ่งจะนำเอาข้อดีของแต่ละเทคนิคมาใช้เพื่อเพิ่มประสิทธิภาพในการทำงานของโปรแกรมโดยรวม

อย่างไรก็ตามการออกแบบโปรแกรมที่คำนวณแบบขนานที่มีประสิทธิภาพต้องคำนึงปัจจัยต่างๆ อันได้แก่ การแบ่งงาน (Partitioning), การสื่อสาร (Communications), การประสานงาน (Synchronization), ความอิสระของข้อมูล (Data Dependencies), การแบ่งงานให้เท่ากัน (Load Balancing), ความละเอียดของงาน (Granularity) เนื่องจากปัจจัยดังกล่าวส่งผลต่อประสิทธิภาพในการทำงานของโปรแกรม เช่น การแบ่งงานที่มีขนาดใหญ่เกินไปและกระจายงานอย่างไม่สม่ำเสมอจะส่งผลให้หน่วยประมวลผลทำงานได้อย่างไม่เต็มที่ และถ้าแบ่งงานที่มีขนาดเล็กเกินไปก็จะส่งผลให้การสื่อสารระหว่างหน่วยประมวลผลมากขึ้น จำนวนของหน่วยประมวลผลที่เพิ่มขึ้นไม่ได้ช่วยในการลดสัดส่วนของงานโดยรวมแต่กลับจะเพิ่มในส่วนของโอเวอร์เฮดในระบบให้มากขึ้น ส่งผลให้ประสิทธิภาพในการทำงานของระบบจึงลดลง การจะเพิ่มประสิทธิภาพในการทำงานของระบบให้มากขึ้นจำเป็นต้องอาศัยวิธีการที่จะมาจัดการกับปัจจัยที่ส่งผลต่อประสิทธิภาพของโปรแกรม เช่น วิธีการแบ่งงานอย่างเหมาะสมและกระจายงานอย่างสม่ำเสมอ การประสานงานและการสื่อสารระหว่างหน่วยประมวลผล ให้สอดคล้องกับความสามารถของคอมพิวเตอร์ที่ใช้งาน เพราะฉะนั้นเมื่อนำโปรแกรมใดๆ ให้ทำงานบนคอมพิวเตอร์ที่มีคุณสมบัติแตกต่างกันจึงจำเป็นต้องมีการปรับแก้ปัจจัยให้สอดคล้องกับ

ระบบด้วย ปัญหาเหล่านี้มีอยู่แล้วในคอมพิวเตอร์แบบขนานที่ใช้เทคโนโลยีอย่างใดอย่างหนึ่ง แต่จะยิ่งมีความซับซ้อนมากขึ้นในคอมพิวเตอร์ขนานแบบลูกผสม

โปรแกรมประยุกต์ประเภทหนึ่งที่มีมักถูกนำมาใช้ในคอมพิวเตอร์สมรรถนะสูงเนื่องจากใช้เวลาและทรัพยากรในการคำนวณมาก ได้แก่ การจำลอง (Simulation) ของระบบต่าง ๆ ในทางวิศวกรรม วัสดุศาสตร์ ชีววิทยา อุตุนิยมวิทยา ฯลฯ ซึ่งในงานลักษณะดังกล่าวมีความซับซ้อนของระบบมากจึงไม่สามารถใช้แบบจำลองเดียวในการคำนวณทั้งหมดได้ ทำให้ต้องมีการใช้แบบจำลองที่เหมาะสมกับระดับต่างๆของข้อมูล ซึ่งจะเรียกว่าแบบจำลองที่มีหลายระดับความละเอียด (Multiscale modeling) [2] ตัวอย่างเช่น แบบจำลองทางฟิสิกส์เพื่อที่จะศึกษาคุณสมบัติของวัตถุก็จะอาศัยข้อมูลจากระดับที่แตกต่างกัน เช่น ระดับควอนตัม (อิเล็กตรอน), ระดับโมเลกุล (อะตอม), ระดับนาโน (กลุ่มของอะตอม) [3] แบบจำลองทางชีววิทยาเพื่อที่จะศึกษาการเกี่ยว กับการทำงานของหัวใจก็จะอาศัยข้อมูลในระดับต่างๆ เช่น ระดับเม็ดเลือดแดง ( $10^{-6}$  m), ระดับหลอดเลือด ( $10^{-3}$  m), ระดับหัวใจ ( $10^{-1}$  m) [4] ซึ่งในแบบจำลองแต่ละระดับของข้อมูลก็มีความสัมพันธ์กันอยู่ ในการนำไปใช้งานจริงของการจำลองโดยใช้แบบจำลองที่มีหลายระดับความละเอียด โดยมากแล้วขนาดของข้อมูลที่นำมาใช้งานก็จะค่อนข้างใหญ่ ในกรณีที่ถูกนำไปคำนวณบนเครื่องคอมพิวเตอร์โดยโปรแกรมการจำลองที่ถูกพัฒนาขึ้นมาเป็นรูปแบบลำดับ เวลาในการคำนวณการจำลองก็จะมาก

ด้วยลักษณะของปัญหาที่มีหลายระดับความละเอียด (Multiscale Problem) ที่พื้นที่สำหรับการคำนวณจะถูกแบ่งออกในลักษณะกริดซึ่งจะถูกคำนวณด้วยฟังก์ชันหนึ่งๆ นอกจากนั้นตรงบริเวณเซลล์ที่สนใจยังสามารถเพิ่มระดับความละเอียดในการคำนวณให้มากขึ้น ได้เป็นพื้นที่สำหรับการคำนวณใหม่ซึ่งอาจจะจะถูกคำนวณด้วยฟังก์ชันที่แตกต่างออกไปและในพื้นที่ย่อยนี้ก็สามารถเพิ่มระดับความละเอียดในการคำนวณบริเวณเซลล์ที่สนใจได้เช่นเดียวกันซึ่งไม่จำเป็นต้องมีขนาดของกริดเท่ากัน โดยเมื่อพิจารณาถึงกระบวนการคำนวณแล้วนั้น ในพื้นที่ระดับความละเอียดหนึ่งๆจะถูกกำหนดให้คำนวณด้วยชุดของหน่วยประมวลผลกลางที่ได้เตรียมการไว้ โดยภายในชุดของหน่วยประมวลผลจะต้องมีการแลกเปลี่ยนข้อมูลกันเพื่อใช้สำหรับการคำนวณ และยังคงต้องมีการแลกเปลี่ยนข้อมูลระหว่างระดับความละเอียดเพื่อที่จะใช้สำหรับการคำนวณต่อไปอีกด้วยดังรูปที่ 1-1 จะพบว่าปัญหาในลักษณะนี้มีความซับซ้อนของรูปแบบของการแบ่งงาน เนื่องจากต้องแบ่งงานตามการคำนวณที่ต่างกันในแต่ละระดับความละเอียด เรียกว่า Functional partitioning และยังคงแบ่งงานในระดับความละเอียดเดียวกันตามข้อมูล เรียกว่า Data partitioning อีกด้วย ดังนั้นวิธีการแบ่งงานอย่างเหมาะสมจึงเป็นเรื่องสำคัญสำหรับปัญหาที่มีหลายระดับความละเอียด โดยเฉพาะอย่างยิ่งบนระบบที่มีความไม่สมมาตรอย่างเช่น คอมพิวเตอร์แบบขนานลูกผสม



รูปที่ 1-1 รูปแบบของปัญหาที่มีหลายระดับความละเอียด

การจำลองสึนามิ (Tsunami Simulation) ก็เป็นปัญหาหนึ่งที่มีหลายระดับความละเอียด กล่าวคือเป็นการจำลองการเกิดคลื่นจากจุดกำเนิดในทะเลโดยอาศัยแบบจำลองทางคณิตศาสตร์ที่ดำเนินการด้วยวิธีการวิเคราะห์เชิงตัวเลข เช่น FDM (Finite Difference Method) ซึ่งการดำเนินการในลักษณะนี้จำเป็นต้องอาศัยข้อมูลของจุดจากบริเวณรอบๆจุดที่กำลังสนใจอยู่มารวมในการคำนวณด้วยเพื่อที่จะทราบถึงความเปลี่ยนแปลงของค่า ณ จุดที่กำลังสนใจอยู่ ซึ่งในการจำลองพื้นที่ที่ถูกนำมาคำนวณจะแบ่งออกเป็นหลายระดับความละเอียดที่ซ้อนกันอยู่ ซึ่งในแต่ละระดับความละเอียดก็จะใช้วิธีการคำนวณที่แตกต่างกันออกไป และความสัมพันธ์ที่เกิดขึ้นระหว่างระดับความละเอียดนั้นก็เกิดขึ้นที่บริเวณขอบของพื้นที่ ในการใช้การจำลองสึนามินั้นข้อมูลพื้นที่ที่นำมาจำลองก็ย่อมที่จะมีขนาดที่ใหญ่ จึงทำให้กว่าจะประมวลผลเสร็จก็เสียเวลาไปมาก ส่งผลให้ไม่บรรลุเป้าหมายของการจำลองสึนามิที่ต้องการจะเตือนภัยในจุดเสี่ยงต่างๆล่วงหน้า และสิ่งที่จะสามารถทำให้การจำลองดังกล่าวบรรลุเป้าหมายได้นั้นก็คือการคำนวณแบบขนาน

งานวิจัยนี้จึงต้องการหาวิธีการจัดการงานของโปรแกรมคำนวณแบบขนาน สำหรับปัญหาที่มีหลายระดับความละเอียดที่ทำงานบนคลัสเตอร์คอมพิวเตอร์ที่มีหน่วยประมวลผลกราฟิกส์ ด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม โดยแบ่งงานและกระจายงานไปยังหน่วยประมวลผลที่ไม่สมมาตรอย่างเหมาะสม และสามารถปรับการจัดการงานให้เหมาะสมกับความสามารถของระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้งาน โดยใช้การจำลองสึนามิเป็นกรณีศึกษา เพื่อที่จะนำวิธีการจัดการงานที่ได้ไปประยุกต์ใช้กับโปรแกรมคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียดอื่นๆ

## 1.2 วัตถุประสงค์ของงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาวิธีการจัดการงานของโปรแกรมคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียดที่ทำงานบนคลัสเตอร์คอมพิวเตอร์ที่มีทั้งหน่วยประมวลผลแบบหลายคอร์และหน่วยประมวลผลกราฟิกส์ด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม โดยสามารถปรับการจัดการงานให้เหมาะสมกับความสามารถของระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้งาน โดยใช้การจำลองสีนามิเป็นกรณีศึกษา เพื่อที่จะนำวิธีการจัดการงานที่ได้ไปประยุกต์ใช้กับโปรแกรมคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียดอื่นๆ

## 1.3 ขอบเขตของการวิจัย

1. เทคนิคการทำงานแบบลูกผสมจะใช้งานเครื่องมือร่วมกันระหว่าง MPICH เวอร์ชัน 1.2.7p1 และ CUDA เวอร์ชัน 4
2. คอมพิวเตอร์แต่ละเครื่องในระบบคลัสเตอร์ที่ใช้ในการทดสอบมีหน่วยประมวลผลแบบ 4 คอร์หนึ่งตัวและหน่วยประมวลผลกราฟิกส์หนึ่งตัว
3. โปรแกรมจำลองสีนามิที่ใช้ทดสอบคือโปรแกรมจำลองสีนามิที่คำนวณแบบลำดับและแบบขนาน ส่วนข้อมูลที่ใช้ทดสอบคือข้อมูลของพื้นที่จริงสำหรับการจำลองสีนามิบริเวณคาบสมุทรอินเดีย

## 1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาหลักการทำงานของโปรแกรมจำลองสีนามิ ทั้งในการคำนวณแบบลำดับและการคำนวณแบบขนาน
2. ศึกษาวิจัยที่เกี่ยวข้องกับการจัดการงานคำนวณแบบขนานลูกผสม และการวิเคราะห์ประสิทธิภาพในการคำนวณ
3. พัฒนาและปรับปรุงโปรแกรมจำลองสีนามิที่ใช้การคำนวณแบบขนาน
4. วางแผนและออกแบบวิธีการจัดการงาน โดยเน้นไปที่วิธีการแบ่งงานและกระจายงานอย่างเหมาะสม
5. นำวิธีการจัดการงานที่ออกแบบมาทดสอบและวัดประสิทธิภาพที่ได้
6. วิเคราะห์ประสิทธิภาพของวิธีการจัดการงานที่ได้รับการทดสอบ
7. สรุปและนำเสนอวิธีที่ใช้ในการจัดการงานคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียด และจัดทำวิทยานิพนธ์

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีจัดการงานของโปรแกรมคำนวณแบบขนาน สำหรับปัญหาที่มีหลายระดับความละเอียด ที่ทำงานบนคลัสเตอร์คอมพิวเตอร์ที่มีหน่วยประมวลผลกราฟิกส์ โดยแบ่งงานและกระจายงานไปยัง หน่วยประมวลผลที่ไม่สมมาตรอย่างเหมาะสม และสามารถปรับการจัดการงานให้เหมาะสมกับ ความสามารถของระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้งาน โดยจะได้รับประสิทธิภาพและความเร็วในการ คำนวณแบบขนานเพิ่มขึ้นจากกรณีศึกษาของการจำลองสึนามิ ซึ่งสามารถนำวิธีการจัดการงานที่ได้ไป ประยุกต์ใช้กับโปรแกรมคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียดอื่นๆได้ รวมไปถึง การใช้ทรัพยากรได้อย่างเต็มประสิทธิภาพมากยิ่งขึ้น

### 1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งออกเป็น 7 บทดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงที่มาและ ความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 ซึ่งกล่าวถึงทฤษฎีพื้นฐานและงานวิจัย ที่เกี่ยวข้อง บทที่ 3 กล่าวถึงกรณีศึกษาที่ใช้ในงานวิจัย บทที่ 4 กล่าวถึงแนวในการทำวิจัย บทที่ 5 กล่าวถึงการทดลองและวิเคราะห์ผลการวิจัย บทที่ 6 กล่าวถึงการประเมินผลการวิจัย และบทที่ 7 กล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

### 1.7 ผลงานที่ได้รับการตีพิมพ์

เนื้อหาบางส่วนของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Application of hybrid parallel programming techniques on tsunami simulation” โดย ธารนิตี เศรษฐพานิชผล และวีระ เหมือนสิน นำเสนอในงานประชุมทางวิชาการ “The 17th International Annual Symposium on Computational Science and Engineering (ANSCSE 17)” ณ มหาวิทยาลัยขอนแก่น จังหวัดขอนแก่น ระหว่างวันที่ 27-29 มีนาคม พ.ศ. 2556

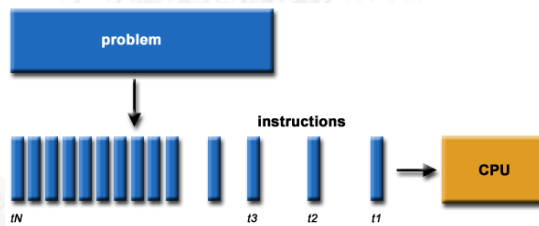
## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

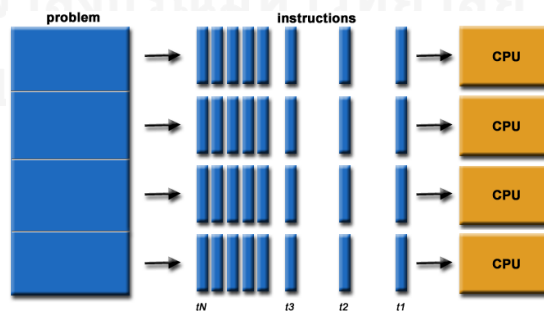
#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 การคำนวณแบบขนาน (Parallel Computing)

เป็นการใช้งานพร้อมกันของหลายทรัพยากรคำนวณเพื่อที่จะแก้ปัญหาทางการคำนวณโดยมีเป้าหมายเพื่อลดเวลาในการทำงาน ซึ่งการทำงานจะแบ่งปัญหาออกเป็นหลายๆส่วนแล้วส่งไปยังหน่วยประมวลผลต่างๆพร้อมกันหลังจากนั้นจึงรวบรวมผลที่ได้จากหน่วยประมวลผลต่างๆกลับคืนมาเพื่อลดเวลารวมของการทำงาน ซึ่งแต่เดิมนั้นโปรแกรมถูกเขียนขึ้นมาเพื่อคำนวณแบบลำดับดังรูปที่ 2-1 และถูกกระจายงานเพื่อคำนวณในรูปแบบขนานดังรูปที่ 2-2



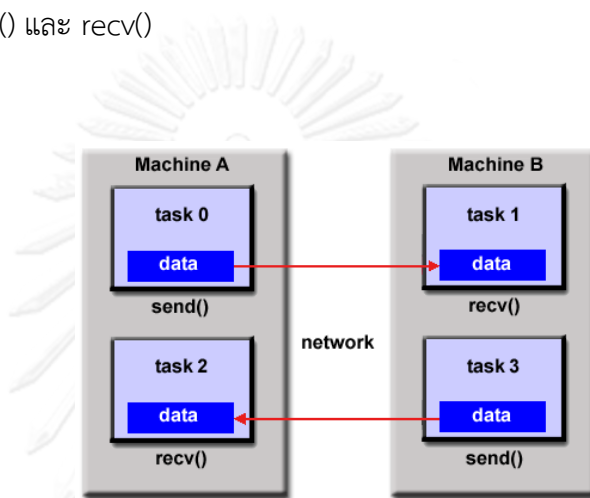
รูปที่ 2-1 รูปแบบการคำนวณแบบลำดับ [1]



รูปที่ 2-2 รูปแบบการคำนวณแบบขนาน [1]

### 2.1.1.1 Message Passing Model

เป็นรูปแบบของการคำนวณแบบขนานซึ่งหน่วยประมวลผลที่มีมากกว่า 1 ตัว ที่อาจจะอยู่บนเครื่องเดียวกันหรือต่างเครื่องกันก็ได้มันได้รับงานที่ถูกแบ่งมาจากการคำนวณแบบลำดับเพื่อนำมาคำนวณ เมื่อต้องมีการสื่อสารเพื่อแลกเปลี่ยนข้อมูลระหว่างกันจะใช้การรับและส่งข้อความในการสื่อสาร โดยจากรูปที่ 2-3 จะพบว่าทั้งเครื่อง A และ B มีงานที่ต้องประมวลผลอยู่เครื่องละ 2 ตัว ซึ่งในการแลกเปลี่ยนข้อมูลระหว่างเครื่อง A และ B นั้นสามารถทำได้ทั้งสองทิศทางคือทั้งไปและกลับโดยผ่านทางคำสั่ง `send()` และ `recv()`



รูปที่ 2-3 ตัวอย่างของการสื่อสารระหว่างหน่วยประมวลผลด้วยการส่งข้อความ [1]

### 2.1.1.2 MPI (Message Passing Interface)

MPI [5] เป็นไลบรารีมาตรฐานสำหรับการเขียนโปรแกรมในรูปแบบ Message Passing คือมีการติดต่อสื่อสารกันระหว่างหน่วยประมวลผลด้วยการส่งข้อความผ่านทางเครือข่าย โดยชุดคำสั่งต่างๆ เช่น

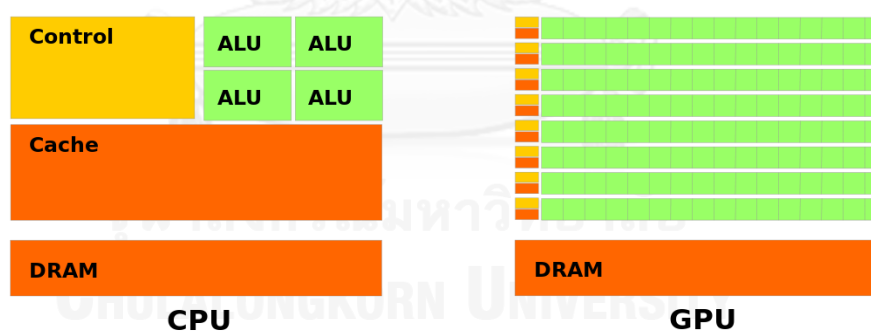
1. `MPI_Init` – กำหนดค่าเบื้องต้นสำหรับการใช้งาน MPI โดยฟังก์ชันนี้จะต้องมีการเรียกใช้งานเป็นอันดับแรกก่อนที่จะมีการใช้งาน MPI ฟังก์ชันอื่นๆ
2. `MPI_Comm_size` – ให้ค่าจำนวนของ Process MPI ที่ถูกกำหนดโดย Communicator
3. `MPI_Comm_rank` – ให้ค่าหมายเลขของหน่วยประมวลผลที่ถูกกำหนดโดย Communicator ซึ่งมีค่าตั้งแต่ 0 ถึง n-1 (จำนวนหน่วยประมวลผลลบหนึ่ง)
4. `MPI_Abort` – ยกเลิกการใช้งาน MPI จะเรียกใช้งานเมื่อมีการเริ่มต้นทำงานของโปรแกรมไม่สมบูรณ์
5. `MPI_Barrier` – ให้หน่วยประมวลผลพักการทำงานจนกว่าจะมีการเรียกใช้งานฟังก์ชันนี้ครบทุกหน่วยประมวลผล



6. MPI\_Wtime – อ่านค่าเวลาของหน่วยประมวลผลที่กำลังใช้งาน MPI
7. MPI\_Isend – ส่งข้อความไปยังปลายทางแบบ non-blocking คือเมื่อส่งข้อความไปยังปลายทางแล้วจะไม่ต้องรอให้การส่งข้อความนั้นเสร็จสิ้นก่อน สามารถทำงานต่อได้ทันที
8. MPI\_Irecv – รอรับข้อความแบบ non-blocking คือขณะกำลังรอรับข้อความจากผู้ติดต่อจะสามารถทำงานอื่นต่อได้ทันทีโดยไม่ต้องรอให้ข้อความมาถึงก่อน
9. MPI\_Wait – รอจนกว่าการส่งหรือรับข้อความของ MPI นั้นจะทำงานเสร็จสิ้น
10. MPI\_Waitall – รอจนกว่าการส่งหรือรับข้อความของทุกงาน MPI นั้นจะทำงานเสร็จสิ้น
11. MPI\_Finalize – สิ้นสุดการใช้งาน MPI

### 2.1.1.3 CUDA (Compute Unified Device Architecture)

โดยปกติแล้วหน่วยประมวลผลกลาง (CPU) จะมีหน้าที่ในการประมวลผลงานหลักและหน่วยประมวลผลกราฟิกส์ (GPU) มีหน้าที่ในการประมวลผลงานกราฟิกส์ แต่ได้ปัจจุบันมีการนำหน่วยประมวลผลกราฟิกส์มาประมวลผลงานทั่วไปเรียกว่า GPGPU (General-Purpose computation on Graphics Processing Units) ในลักษณะที่เป็นหน่วยประมวลผลแบบขนานประเภทหนึ่งโดยมีสถาปัตยกรรมดังในรูปที่ 2-4



รูปที่ 2-4 เปรียบเทียบสถาปัตยกรรมระหว่าง CPU และ GPU [6]

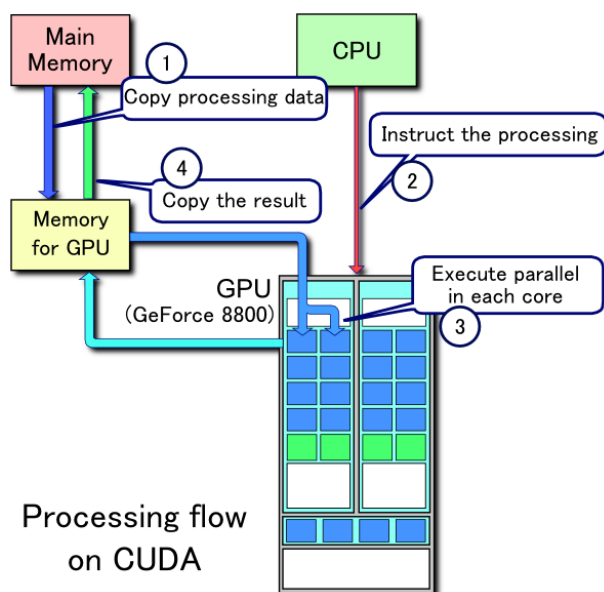
CUDA เป็นแพลตฟอร์มการคำนวณแบบขนาน (Parallel Computing Platform) และรูปแบบการเขียนโปรแกรม (Programming model) ที่ถูกสร้างขึ้นโดยบริษัท NVIDIA [6] ซึ่งจะเพิ่มประสิทธิภาพให้การคำนวณงานด้วยการควบคุมหน่วยประมวลผลกราฟิกส์ (GPU) อันจะเป็นการช่วยแบ่งเบาภาระของหน่วยประมวลผลกลาง (CPU) ลงได้ CUDA ใช้งานได้กับหลายภาษา เช่น C, C++, Fortran เป็นต้น

ในการประมวลผลทุกครั้งของ CUDA จะมีขั้นตอนตามรูปที่ 2-5 ที่สามารถสรุปได้ดังต่อไปนี้

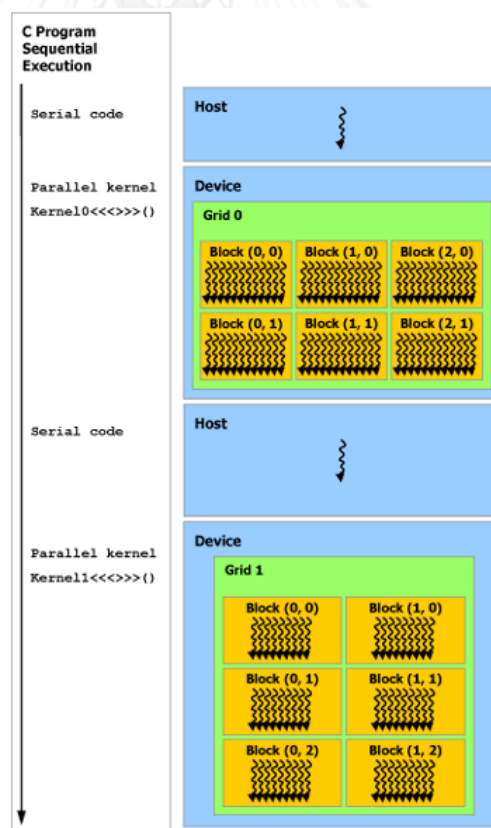
1. คัดลอกข้อมูลที่จะประมวลผลจากหน่วยความจำหลักไปยังหน่วยความจำของกราฟิกส์การ์ด
2. หน่วยประมวลผลกลางสั่งการประมวลผลบนหน่วยประมวลผลกราฟิกส์
3. หน่วยประมวลผลกราฟิกส์ดำเนินการประมวลผลแบบขนานในแต่ละแกน (Core)
4. คัดลอกผลที่ได้จากการประมวลผลจากหน่วยความจำของหน่วยประมวลผลกราฟิกส์การ์ดไปยังหน่วยความจำหลัก

หลักการทำงานของ CUDA สามารถแสดงให้เห็นได้ดังรูปที่ 2-6 จากภาพด้านซ้ายคือลำดับการดำเนินการของโปรแกรมที่เขียนด้วยภาษาซี ส่วนด้านขวาจะอธิบายการทำงานของหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์ให้เป็นรูปธรรม จากภาพทางด้านขวาคือความที่เขียนว่า “Host” จะแสดงถึงการทำงานบนหน่วยประมวลผลกลาง และข้อความที่เขียนว่า “Device” จะแสดงถึงการทำงานบนหน่วยประมวลผลกราฟิกส์ และลูกศรแต่ละอันนั้นจะหมายถึงงานหนึ่งงาน จะพบว่าหน่วยประมวลผลกลางนั้นสามารถประมวลผลได้ที่ละหนึ่งงาน แต่บนหน่วยประมวลผลกราฟิกส์สามารถประมวลผลพร้อมกันได้เป็นจำนวนมาก โดยงานที่จะแบ่งเพื่อทำงานบนหน่วยประมวลผลกราฟิกส์นั้นจะสามารถแบ่งโดยอาศัยหลักการดังรูปที่ 2-5

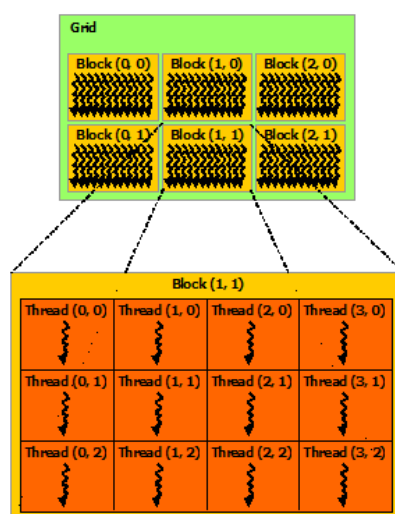
จากรูปที่ 2-7 จะแสดงถึงโครงสร้างของหน่วยประมวลผลบนหน่วยประมวลผลกราฟิกส์ ซึ่งแต่ละฟังก์ชันที่จะต้องดำเนินการบนหน่วยประมวลผลกราฟิกส์นั้นจะต้องการ Grid อย่างน้อย 1 Grid ในการทำงานโดยภายใน Grid จะประกอบไปด้วย Block ตามจำนวนที่กำหนดและในแต่ละ Block จะประกอบไปด้วย Thread ตามจำนวนที่กำหนดเช่นเดียวกัน ซึ่งในแต่ละ Thread นั้นจะสามารถประมวลงานได้ 1 งาน ซึ่งตำแหน่งของ Block และ Thread จะจัดเรียงอยู่ในลักษณะสองมิติ



รูปที่ 2-5 ขั้นตอนการประมวลผลบน CUDA [6]



รูปที่ 2-6 หลักการทำงานของ CUDA [6]



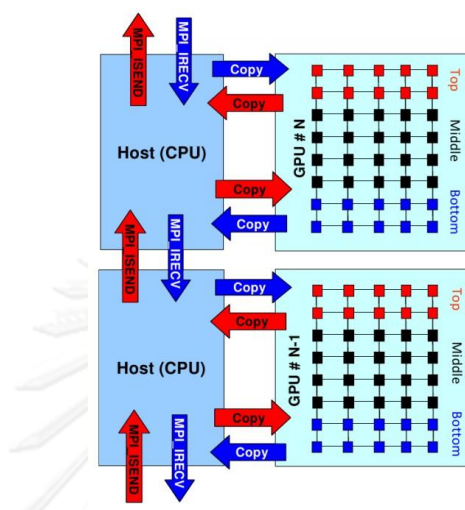
รูปที่ 2-7 โครงสร้างของหน่วยประมวลผลบนหน่วยประมวลผลกราฟิกส์ [6]

#### 2.1.1.4 Hybrid Parallel Programming

การเขียนโปรแกรมแบบลูกผสมเกิดจากสถาปัตยกรรมของคอมพิวเตอร์แบบขนานที่มีการประมวลผลแบบขนานในหลายระดับ โดยเฉพาะในระดับที่ควบคุมได้โดยโปรแกรมเมอร์ ในปัจจุบันคอมพิวเตอร์แบบขนานที่มีขนาดใหญ่มักเป็นคลัสเตอร์คอมพิวเตอร์เนื่องจากเป็นสถาปัตยกรรมที่มีความสามารถในการขยายระบบสูง ในขณะเดียวกันคอมพิวเตอร์แต่ละเครื่องในคลัสเตอร์ก็ยังเป็นคอมพิวเตอร์แบบขนานโดยใช้สถาปัตยกรรมแบบ multicore/multiprocessor และยังสามารถเพิ่มความสามารถในการคำนวณด้วย GPU ทั้งนี้สถาปัตยกรรมทั้ง 3 แบบ ต่างก็มีวิธีการเขียนโปรแกรมและ run-time system ที่แตกต่างกันสำหรับคลัสเตอร์เป็นสถาปัตยกรรมแบบ distributed memory ที่มีรูปแบบการเขียนโปรแกรมแบบ message-passing และมีไลบรารีที่นิยมใช้กันคือ MPI (Message Passing Library) multicore/ multiprocessor เป็นสถาปัตยกรรมแบบ symmetric multiprocessor (SMP) ที่มีรูปแบบการเขียนโปรแกรมแบบ thread programming และมีไลบรารีที่นิยมใช้กันคือ OpenMP และ Pthreads ส่วน GPU เป็นสถาปัตยกรรมแบบ Single Instruction Multiple Data (SIMD) ที่มีรูปแบบการเขียนโปรแกรมแบบ data parallel programming และมีไลบรารีที่นิยมใช้กันคือ CUDA และ OpenCL

การเขียนโปรแกรมแบบขนานลูกผสมเป็นการรวมของเทคนิคการเขียนโปรแกรมแบบขนานมากกว่าหนึ่งแบบ ซึ่งนำเอาข้อดีของแต่ละเทคนิคมาใช้เพื่อเพิ่มประสิทธิภาพในการทำงานของโปรแกรมโดยรวม [7, 8] ซึ่งในงานวิจัยนี้จะนำ MPI ซึ่งมีประโยชน์ในการติดต่อสื่อสารกับระบบคลัสเตอร์มารวมกับ CUDA ซึ่งมีประโยชน์ในการใช้งานกราฟิกส์การ์ดสำหรับคำนวณงานประมวลผล

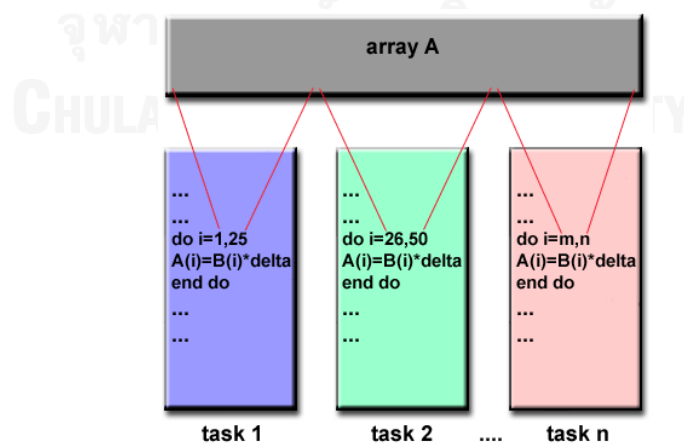
ทั่วไป ดังนั้นการรวมกันของเทคนิคการเขียนโปรแกรมทั้งสองแบบจึงส่งผลให้บนระบบคลัสเตอร์มีทรัพยากรที่สามารถรองรับงานประมวลผลได้มากขึ้น



รูปที่ 2-8 ตัวอย่างการทำงานของโปรแกรมแบบขนานลูกผสม [7, 8]

### 2.1.1.5 Data-Parallel model

เป็นรูปแบบที่เน้นการคำนวณแบบเดียวกันบนชุดข้อมูลที่อยู่ในรูปแบบของอาร์เรย์ โดยสามารถแบ่งการดำเนินการกับข้อมูลออกเป็นส่วนๆได้ ดังรูปที่ 2-9



รูปที่ 2-9 รูปแบบของ Data-Parallel model [1]

### 2.1.1.6 Asymmetric Multi Processors

โดยทั่วไปคอมพิวเตอร์แบบขนานมักใช้หน่วยประมวลผลที่มีประสิทธิภาพเท่า ๆ กัน เช่น Symmetric Multi-Processor (SMP), multicore processor และ คลัสเตอร์ ซึ่งทำให้ง่ายต่อการแบ่งงานให้เท่ากัน (load balancing) รวมถึงการวิเคราะห์และปรับปรุงประสิทธิภาพด้วยวิธีต่าง ๆ แต่ในปัจจุบัน มีแนวคิดของการใช้หน่วยประมวลผลที่มีประสิทธิภาพไม่เท่ากันซึ่งอาจทำให้ประสิทธิภาพโดยรวมของระบบดีกว่า เช่น asymmetric multicore processor [9] และการใช้ CPU ร่วมกับ GPU เป็นต้น ซึ่งทำให้การแบ่งงาน รวมถึงการวิเคราะห์และปรับปรุงประสิทธิภาพด้วยวิธีต่าง ๆ มีความซับซ้อนมากขึ้น

### 2.1.1.7 ปัญหาที่มีหลายระดับความละเอียด (Multi-scale Problem)

ในงานทางด้านวิศวกรรม และวิทยาศาสตร์ การหาค่าตอบของเหตุการณ์จำลองในแต่ละกรณีนั้น เนื่องจากตัวปัญหาที่ค่อนข้างมีความซับซ้อนของระบบมากจึงไม่สามารถใช้แบบจำลองเดียวในการคำนวณทั้งหมดได้ ทำให้ต้องมีการใช้แบบจำลองที่เหมาะสมกับระดับต่างๆของข้อมูล ซึ่งเป็นปัญหาที่มีหลายระดับความละเอียด และซึ่งในแต่ละระดับความละเอียดของข้อมูลก็จะมีความสัมพันธ์กันอยู่ในบริเวณขอบของพื้นที่ ดังเช่นในงานวิจัยเรื่อง Performance Characteristics of the Multi-Zone NAS Parallel Benchmarks [10] ซึ่งได้กล่าวถึงลักษณะของการพัฒนาโปรแกรมแบบขนานลูกผสมในรูปแบบต่างๆเพื่อจัดการกับปัญหาที่มีหลายระดับความละเอียดได้อย่างมีประสิทธิภาพ

## 2.1.2 การปรับแต่งประสิทธิภาพของโปรแกรมคำนวณแบบขนาน (Performance Tuning of Parallel Program)

ในการที่จะปรับแต่งประสิทธิภาพของโปรแกรมแบบขนานนั้นต้องทราบถึงปัจจัยที่ส่งผลต่อประสิทธิภาพในการคำนวณเสียก่อน ตัวอย่างเช่น การแบ่งข้อมูลที่จะใช้สำหรับการประมวลผล โดยเน้นไปที่การแบ่งงานและกระจายงานไปยังหน่วยประมวลผลอย่างสม่ำเสมอ เพื่อลดปริมาณการสื่อสารและการประสานงานระหว่างหน่วยประมวลผล ต้องทำให้เหมาะสมกันขนาดของปัญหา

### 2.1.2.1 การแบ่งงานและกระจายงาน (Partitioning and Load Balancing)

ในการคำนวณแบบขนาน การแบ่งข้อมูลที่ใช้สำหรับการประมวลผลออกเป็นส่วนๆนั้นคือสิ่งที่จำเป็นต้องทำเพื่อที่จะให้แต่ละหน่วยประมวลผลนั้นทำการคำนวณไปพร้อมๆกัน โดยวิธีการแบ่งงานนั้นสามารถทำได้ทั้งในรูปแบบคัมหลักและตามแถว โดยในงานวิจัยนี้จะเลือกใช้การแบ่งงานตามแถวเนื่องจากสามารถนำข้อมูลมาเชื่อมต่อกันได้ง่ายและรวดเร็วกว่าการแบ่งข้อมูลตามหลัก

โดยรูปแบบของการแบ่งงานที่จะสามารถทำให้การกระจายงานได้อย่างสม่ำเสมอ ต้องคำนึงถึงปัจจัยที่ส่งผลต่อความเร็วในการคำนวณ และความซับซ้อนของข้อมูล เช่น ความละเอียดของข้อมูลที่แตกต่างกันในแต่ละพื้นที่ จำนวนรอบที่ได้ข้อมูลนั้นๆจะถูกประมวลผล โดยในงานวิจัยนี้จำเป็นต้องหาความสัมพันธ์ของปัจจัยต่างๆของโปรแกรมจำลองสึนามิ ซึ่งใช้การแบ่งงานตามขนาดพื้นที่ของการคำนวณ ซึ่งแบ่งให้แต่ละหน่วยประมวลผลกลางทำงานเท่าๆกัน และแบ่งให้ตามอัตราส่วนของประสิทธิภาพในการทำงานของหน่วยประมวลผลกราฟิกส์เทียบกับหน่วยประมวลผลกลาง โดยจะต้องหาอัตราส่วนการแบ่งงานระหว่างการจำลองระยะใกล้และการจำลองระยะไกลออกมาเป็นวิธีจัดการงานสำหรับปัญหาที่มีความซับซ้อนในการคำนวณ

### 2.1.2.2 Zero-Copy Memory

Zero-Copy Memory เป็นเทคนิคสำหรับการสื่อสารระหว่างหน่วยประมวลผลกลางกับหน่วยประมวลผลกราฟิกส์ โดยเทรตของหน่วยประมวลผลกราฟิกส์สามารถเข้าถึงหน่วยความจำหลักได้โดยตรงโดยไม่ต้องมีการเรียกใช้คำสั่งเพื่อทำการคัดลอกข้อมูลแต่อย่างใด ในการใช้งานเทคนิคนี้หน่วยความจำของโฮสจำเป็นต้องจองพื้นที่แบบ Page-locked memory โดยทำการเรียกใช้งานคำสั่ง `cudaHostAlloc()` ตามด้วยพารามิเตอร์ `cudaHostAllocMapped` และการที่จะให้ Device Pointer สามารถใช้ Page-locked memory จำเป็นต้องมีการเรียกใช้คำสั่ง `cudaHostGetDevicePointer()` ก่อนที่จะมีการเรียกใช้เคอเนล อย่างไรก็ตามการใช้เทคนิคนี้ก็มีข้อจำกัดอยู่บางประการ โดยที่เวอร์ชันของกราฟิกส์การ์ดที่ใช้ต้องมากกว่า 1.0 และสามารถใช้งานหน่วยความจำได้สูงสุด 4 GB

### 2.1.3 การวัดประสิทธิภาพของโปรแกรมคำนวณแบบขนาน (Performance Measuring of Parallel Program)

ในการคำนวณแบบขนานจำเป็นที่จะต้องมีการแสดงให้เห็นถึงประสิทธิภาพที่เพิ่มขึ้นหรือลดลงจากการเพิ่มขึ้นของจำนวนหน่วยประมวลผลออกมาในรูปของตัวเลขเพื่อที่จะจะสามารถเปรียบเทียบได้อย่างสะดวก

#### 2.1.3.1 ความเร็วที่เพิ่มขึ้น (Speedup)

ในการทดสอบประสิทธิภาพของการคำนวณแบบขนานจำเป็นต้องมีสิ่งที่สามารถแสดงถึงความสามารถในการคำนวณที่เพิ่มมากขึ้นจากการคำนวณแบบลำดับ ซึ่งความเร็วที่เพิ่มขึ้นจากการคำนวณแบบขนานสามารถคำนวณได้จาก

$$S_p = \frac{T_s}{T_p} = \frac{\text{Sequential\_time}}{\text{Patallel\_time}}$$

- p จำนวนของหน่วยประมวลผลที่ใช้ในการคำนวณ
- $S_p$  ความเร็วที่เพิ่มขึ้นจากการคำนวณแบบขนานเมื่อใช้ p หน่วยประมวลผล
- $T_s$  เวลาที่ได้จากการคำนวณแบบลำดับ 1 หน่วยประมวลผล
- $T_p$  เวลาที่ได้จากการคำนวณแบบขนาน p หน่วยประมวลผล

ซึ่งในอุดมคติแล้ว ความเร็วที่เพิ่มขึ้นควรมีค่าเท่ากับจำนวนของหน่วยประมวลผลที่ใช้ในการคำนวณ ( $S_p = p$ ) [11]

### 2.1.3.2 ประสิทธิภาพของการทำงาน (Efficiency)

เมื่อทำการคำนวณแบบขนานแล้วก็จะสามารถหาความเร็วที่เพิ่มขึ้นได้จากการใช้หน่วยประมวลผลตามจำนวนที่กำหนด ดังนั้นจึงมีการกำหนดค่าเพื่อที่จะสามารถแสดงถึงความคุ้มค่าของการใช้งานจำนวนของหน่วยประมวลผลนั้นๆ ซึ่งประสิทธิภาพที่ได้รับจากการเพิ่มจำนวนของหน่วยประมวลผลสามารถหาได้จาก [12]

$$E_p = \frac{S_p}{p} = \frac{T_s}{pT_p}$$

- p จำนวนของหน่วยประมวลผลที่ใช้ในการคำนวณ
- $E_p$  ประสิทธิภาพที่ได้รับจากการเพิ่มจำนวนของหน่วยประมวลผล
- $S_p$  ความเร็วที่เพิ่มขึ้นจากการคำนวณแบบขนานเมื่อใช้ p หน่วยประมวลผล
- $T_s$  เวลาที่ได้จากการคำนวณแบบลำดับ 1 หน่วยประมวลผล
- $T_p$  เวลาที่ได้จากการคำนวณแบบขนาน p หน่วยประมวลผล



### 2.1.4 การเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนาน (Program Profiling)

การเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนานนี้ เป็นการจัดเก็บข้อมูลเวลาที่ใช้ในการทำงานให้อยู่ในเชิงของสถิติ เพื่อนำมาใช้ในการวิเคราะห์ประสิทธิภาพการทำงานของแต่ละส่วนการทำงานในโปรแกรมซึ่งจะช่วยในการปรับปรุงประสิทธิภาพของโปรแกรมได้ โดยในงานวิจัยนี้ได้ใช้โปรแกรมเก็บข้อมูลการทำงานของโปรแกรกดังต่อไปนี้

#### 2.1.4.1 IPM Version 0.9.80 (Integrated Performance Monitoring) and Ploticus Version 2.4.1

IPM [13] เป็นเครื่องมือเก็บข้อมูลเวลาการทำงานของโปรแกรมแบบขนาน ซึ่งมีขนาดเล็ก และสามารถติดตั้งเพื่อใช้งานได้ง่าย จะเกิดโอเวอร์เฮดเพียงเล็กน้อยจากการที่ใช้เครื่องมือนี้เพื่อเก็บข้อมูลการทำงานของโปรแกรมแบบขนาน โดยข้อมูลที่โปรแกรมมุ่งเป้าไปได้แก่ การสื่อสาร การประมวลผล และข้อมูลเข้า-ออก ซึ่งผลลัพธ์จากการเก็บข้อมูลของการทำงานแบบขนานจะออกมาเป็น XML ไฟล์ และเพื่อให้ง่ายต่อการวิเคราะห์ โปรแกรม Ploticus [14] สามารถที่จะแปลงข้อมูล XML ดังกล่าวให้เป็นกราฟิกอยู่ใน HTML ไฟล์

นอกจากการใช้เครื่องมือ IPM เพื่อเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนานแล้ว ก็จะมีการแทรกคำสั่ง MPI\_Wtime ลงในโปรแกรม เพื่อได้เวลาเริ่มต้นและสิ้นสุดของโปรแกรม ผลต่างของเวลาดังกล่าวก็จะเป็นเวลาที่ใช้ในการทำงานของส่วนนั้นๆ และแทรกคำสั่ง cudaEventRecord ลงในโปรแกรมส่วนที่มีการทำงานของหน่วยประมวลผลกราฟิกส์ เพื่อได้เวลาเริ่มต้นและสิ้นสุด เพื่อหาเวลาที่ใช้ในการทำงานของแต่ละส่วนบนหน่วยประมวลผลกราฟิกส์ เช่นเดียวกัน ซึ่งการเก็บข้อมูลการทำงานของโปรแกรมทั้งหมดนั้นก็เพื่อนำไปวิเคราะห์และปรับปรุงประสิทธิภาพของโปรแกรม

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่มีหลายระดับความละเอียด : กรณีศึกษาการจำลองสึนามิ (Parallel program development with Adaptive Performance Tuning for Multi-Scale problem : A case study of tsunami simulation)

การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่มีหลายระดับความละเอียดนี้เป็นผลงานวิจัยของนายสิทธิกร ถาวรรัตนวิช นิสิตภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งผศ.ดร.วีระ เหมืองสิน เป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก และผศ.ดร.อาณัติ เรืองรัมย์ เป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ในปี 2553 [15] โดยใน

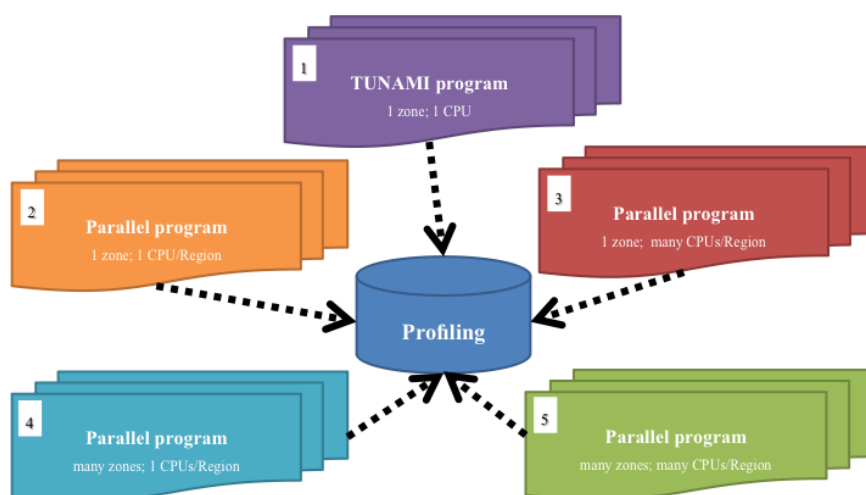
ปัญหาของงานวิจัยนี้เกิดขึ้นมาจากการจำลองสึนามิเป็นปัญหาที่มีหลายระดับความละเอียด เมื่อนำมาพัฒนาเป็นโปรแกรมที่คำนวณแบบขนานจะมีความซับซ้อนในการออกแบบโปรแกรมให้มีประสิทธิภาพที่เพิ่มขึ้น ดังนั้นมีการวิจัยเพื่อค้นหาวิธีการจัดการงานสำหรับโปรแกรมคำนวณแบบขนานจากกรณีศึกษา ด้วยการเก็บข้อมูลการทำงานของโปรแกรมในแต่ละส่วนการทำงานด้วยข้อมูลตัวอย่าง จากนั้นหาความสัมพันธ์ของประสิทธิภาพเชิงเวลากับรูปแบบในการแบ่งงาน ขนาดของปัญหา และโอเวอร์เฮดต่างๆที่เกิดขึ้นระหว่างการติดต่อสื่อสารระหว่างหน่วยประมวลผล เพื่อให้โปรแกรมสามารถทำงานได้อย่างเต็มประสิทธิภาพ และใช้เวลาในการคำนวณรวมลดลงอันเนื่องมาจากการกระจายงานอย่างสม่ำเสมอให้เหมาะสมกับขนานของปัญหาและขนาดของระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้ งาน อีกทั้งยังสามารถปรับเปลี่ยนตามข้อกำหนดในเรื่องของเวลาที่ใช้ และข้อกำหนดของจำนวนหน่วยประมวลผลที่ใช้อีกด้วย

ซึ่งในงานวิจัยนี้จะทำการเก็บข้อมูลการทำงานของโปรแกรมที่คำนวณแบบขนานด้วยโปรแกรม IPM (Integrated Performance Monitoring) [13] และ MpiP (Lightweight, Scalable MPI Profiling) [16] เพื่อนำมาวิเคราะห์ข้อมูลที่เกี่ยวข้องกับประสิทธิภาพในการทำงานของการคำนวณแบบขนานระบบคลัสเตอร์คอมพิวเตอร์ ซึ่งทำการทดสอบบนระบบคลัสเตอร์ TERA ของ Thai National Grid Center มหาวิทยาลัยเกษตรศาสตร์ ประเทศไทย และระบบคลัสเตอร์ TSUBAME ของ GSIC Center สถาบันเทคโนโลยีแห่งโตเกียว (Tokyo Institute of Technology / TiTech) ประเทศญี่ปุ่น

แนวทางในการทดสอบของงานวิจัยนี้จะประกอบไปด้วยการทดสอบทั้งหมด 5 ส่วน ซึ่งได้ใช้ข้อกำหนดในเรื่องของระยะเวลาที่ใช้ในการคำนวณ และจำนวนของหน่วยประมวลผลที่ใช้คำนวณบนระบบคลัสเตอร์คอมพิวเตอร์ทั้งสองระบบ ซึ่งทำให้ทราบระยะเวลาที่ใช้ในการคำนวณจริงของแต่ละส่วนการทำงานและทราบระยะเวลาของโอเวอร์เฮดที่ใช้ในการติดต่อสื่อสารระหว่างหน่วยประมวลผล ซึ่งมีรายละเอียดและปรากฏดังรูปที่ 2-10 ดังนี้

1. ทดสอบด้วยโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับที่ใช้การคำนวณหนึ่งโชน ด้วยจำนวนหน่วยประมวลผลหนึ่งหน่วย
2. ทดสอบด้วยโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานที่ใช้การคำนวณหนึ่งโชน และใช้การแบ่งงานโดยใช้จำนวนหน่วยประมวลผลหนึ่งหน่วยต่อหนึ่งพื้นที่ย่อย
3. ทดสอบด้วยโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานที่ใช้การคำนวณหนึ่งโชน แต่ใช้การแบ่งงานโดยใช้จำนวนหน่วยประมวลผลมากกว่าหนึ่งหน่วยต่อหนึ่งพื้นที่ย่อย
4. ทดสอบด้วยโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานที่ใช้การคำนวณหลายโชน พร้อมกัน และใช้การแบ่งงานโดยใช้จำนวนหน่วยประมวลผลหนึ่งหน่วยต่อหนึ่งพื้นที่ย่อย

5. ทดสอบด้วยโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานที่ใช้การคำนวณหลายโชนพร้อมกัน แต่ใช้การแบ่งงานโดยใช้จำนวนหน่วยประมวลผลมากกว่าหนึ่งหน่วยต่อหนึ่งพื้นที่ย่อย



รูปที่ 2-10 การทดสอบเพื่อใช้ในการวิเคราะห์ข้อมูล

สำหรับข้อมูลที่น่ามาวิจัย ได้นำข้อมูลสำหรับการจำลองสึนามิมาบางส่วนโดยเลือกให้ครอบคลุมการเปลี่ยนแปลงของทุกระดับความละเอียดจากตารางที่ 3-1 ซึ่งจะมีรายละเอียดดังตารางที่ 2-1

ตารางที่ 2-1 พิกัดและขนาดของพื้นที่ในแต่ละระดับความละเอียดที่ใช้ในการทดสอบ  
สำหรับโปรแกรมจำลองสึนามิที่คำนวณแบบขนาน

Region levels	Region ID	No. of Grid		Grid size (units)	Sea area (units)	The coordinate of grid on overlapped region			
						From		To	
		X	Y			X	Y	X	Y
R1	1	690	840	579,600	449,678	-	-	-	-
R2	21	721	841	606,361	547,045	271	481	360	585
	22	721	841	606,361	496,667	271	587	360	691

R3	211	241	325	78,325	67,620	482	424	561	531
	212	637	469	298,753	275,475	350	612	561	767
	221	639	487	311,193	234,716	430	6	642	167
R4	2111	487	325	158,275	131,692	68	68	229	175
	2112	487	421	205,027	179,132	68	180	229	319
	2121	643	469	301,567	248,562	400	56	613	211
	2211	703	457	321,271	182,585	403	330	636	481

โดยการวิเคราะห์การทำงานในส่วนต่างๆเพื่อนำไปสู่วิธีการจัดการงานจะเป็นไปดังนี้

1. การวิเคราะห์ประสิทธิภาพเชิงเวลาของแต่ละส่วนการทำงานของโปรแกรม
  - 1.1. การวิเคราะห์โปรแกรมจำลองสีนามิตด้วยการคำนวณแบบลำดับ
  - 1.2. การวิเคราะห์โปรแกรมจำลองสีนามิตด้วยการคำนวณแบบขนาน
2. การวิเคราะห์ประสิทธิภาพเชิงเวลาที่ขึ้นกับขนาดของปัญหา
3. การวิเคราะห์การติดต่อสื่อสารระหว่างหน่วยประมวลผล
  - 3.1. การแลกเปลี่ยนข้อมูลภายนอกของแต่ละระดับความละเอียด
  - 3.2. การแลกเปลี่ยนข้อมูลภายในของแต่ละระดับความละเอียด
  - 3.3. การแลกเปลี่ยนข้อมูลภายนอกของแต่ละระดับความละเอียด เมื่อมีการแบ่งงานในแต่ละระดับความละเอียด
  - 3.4. การรอรับข้อมูลของการติดต่อสื่อสารระหว่างหน่วยประมวลผล

ซึ่งผลของการวิเคราะห์ข้อมูลเวลาการทำงานในแต่ละส่วนของโปรแกรมก็จะได้ตารางซึ่งประกอบไปด้วยค่าของตัวแปรในสมการเชิงเส้นสำหรับการประมาณเวลาที่ใช้ของส่วนการทำงานต่างๆ จากนั้นจึงนำไปสู่การหาค่า Threshold ที่ใช้สำหรับการแบ่งขนาดปัญหาที่ต้องการให้แต่ละหน่วยประมวลผล และทำการคำนวณเวลาในส่วนการทำงานต่างๆด้วยสมการของการทำงานนั้นๆ เพื่อประมาณค่าเวลาที่จะใช้ในการทำงานของโปรแกรม โดยการหาค่า Threshold สามารถคำนวณได้จากขนาดของปัญหาทั้งหมดหารด้วยจำนวนของหน่วยประมวลผลทั้งหมดที่ใช้ในการทำงาน โดยจำนวนของหน่วยประมวลผลต้องไม่ถูกกำหนดมากหรือน้อยเกินไป

$$\text{Threshold} = \frac{AsR1 + AsR2 + AsR3 + AsR4 + \dots + AsRn}{\text{Number of processor}}$$

จากนั้นจึงนำค่า Threshold ที่ได้มาหาจำนวนการแบ่งปัญหาในแต่ละระดับความละเอียด โดยนำขนาดของปัญหาในแต่ละระดับความละเอียดมาหารด้วยค่า Threshold เพื่อให้ได้จำนวนหน่วยประมวลผลต่อระดับความละเอียด (p)

$$p = \frac{AsRn}{\text{Threshold}}$$

จากนั้นทำการหาขนาดของปัญหาที่ถูกแบ่งตามแนวนอน (Ap) ด้วยสมการ

$$Ap = \text{lenX} * \frac{\text{lenY}}{p}$$

จากนั้นก็คำนวณหาปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสาร (B) เพื่อนำไปใช้ในสมการเชิงเส้นสำหรับประมาณเวลาที่ใช้ในการติดต่อสื่อสาร

$$B = \text{length} * V\_size * \text{Timestep} * \text{Row}$$

ผลจากการที่มีวิธีการจัดการงานที่คำนวณแบบขนานอย่างเหมาะสมทำให้สามารถกำหนดจำนวนของการแบ่งงานในแต่ละระดับความละเอียดเพื่อกระจายงานไปยังหน่วยประมวลผลได้อย่างเหมาะสม และให้สามารถทำได้อย่างได้อย่างมีประสิทธิภาพ ดังนั้นวิธีการจัดการงานที่คำนวณแบบขนาน

ดังกล่าวก็สามารถนำไปประยุกต์เพิ่มเติมกับระบบที่คำนวณแบบขนานอื่นๆได้ แต่ในกรณีที่หน่วยประมวลผลแต่ละตัวนั้นมีความสามารถไม่เท่ากัน วิธีการจัดการงานที่คำนวณแบบขนานดังกล่าวยังไม่สามารถแบ่งงานเพื่อกระจายงานอย่างเหมาะสมได้ จึงควรมีวิธีที่จะมาจัดการงานที่คำนวณแบบขนานสำหรับระบบที่มีความสามารถของหน่วยประมวลผลแต่ละตัวไม่เท่าเทียมกัน

## 2.2.2 ระบบเตือนภัยสึนามิด้วยการจำลองแบบทันกาลบนหน่วยประมวลผลกราฟิกส์ (Tsunami Warning System with Real-Time Simulation on Graphics Processing Unit)

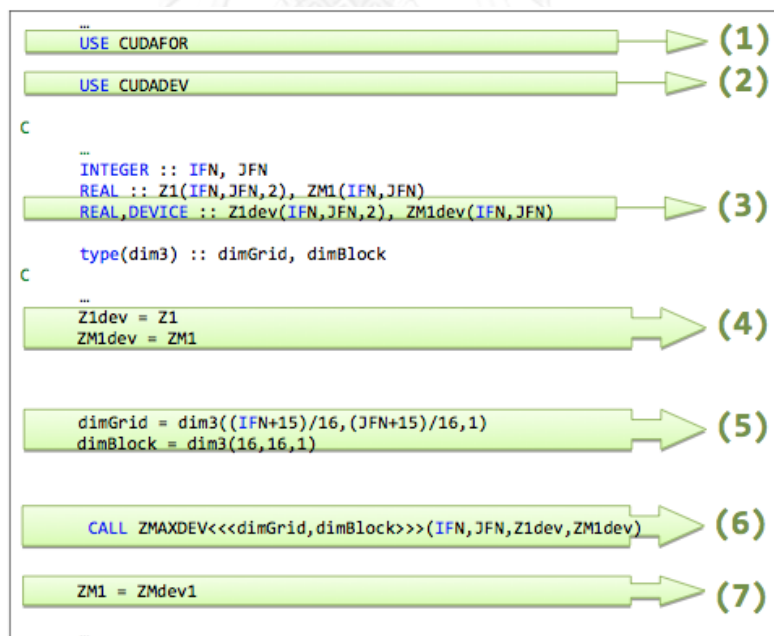
ระบบเตือนภัยสึนามิด้วยการจำลองแบบทันกาลบนหน่วยประมวลผลกราฟิกส์ได้รับการพัฒนาในโครงการทางวิศวกรรมของ นายเชม อำนวยลาภ และนายฉัตรชัย พิรพัฒน์ดิษฐ์ นิสิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งเป็นผู้พัฒนา และผศ.ดร.วีระ เหมืองสิน เป็นอาจารย์ที่ปรึกษาโครงการ ในปี 2553 [17] โดยปัญหาของโครงการนี้เกิดขึ้นมาจากการที่โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับไม่สามารถนำไปใช้งานแบบทันกาล (Real-time) ได้ต้องอาศัยการคำนวณกรณีที่จะเกิดขึ้นล่วงหน้าซึ่งให้ผลลัพธ์ที่ไม่มีความแม่นยำเมื่อเทียบกับการคำนวณโดยใช้ข้อมูลจากกรณีที่เกิดขึ้นจริง ดังนั้นในโครงการนี้จึงได้ปรับปรุงระบบจำลองสึนามิให้สามารถคำนวณแบบขนานเพื่อให้สามารถใช้งานแบบเรียลไทม์ โดยนำสถาปัตยกรรม CUDA มาใช้ในการเขียนชุดคำสั่งเพื่อให้ทำงานแบบขนานบนหน่วยประมวลผลกราฟิกส์ ซึ่งโปรแกรมจำลองสึนามิจะถูกพัฒนาโดยใช้คอมไพเลอร์ CUDA Fortran ซึ่งพัฒนาร่วมกันโดยบริษัท Portland Group, Inc. (PGI) และบริษัท NVIDIA ทำให้สามารถเรียกใช้งานชุดคำสั่ง CUDA ผ่านภาษา Fortran (CUDA Fortran) ได้โดยตรง ซึ่งคอมไพเลอร์ดังกล่าวได้ออกเวอร์ชันเสริมที่สามารถผนวกเข้ากับ Visual Studio ได้ทั้งในรุ่น 32 และ 64 บิต

โดยแนวทางในการแก้ปัญหาของโครงการนี้ก็จะเป็นการวิเคราะห์โครงสร้างของโปรแกรมจำลองสึนามิเดิมเพื่อที่จะทำการปรับปรุงให้สามารถทำงานบนหน่วยประมวลผลกราฟิกส์ได้ ซึ่งจะแบ่งออกเป็น 2 ขั้นตอนหลักได้แก่

1. การแก้ไขส่วนของการทำงานเดิมซึ่งเป็นจำนวนรอบให้สามารถทำงานบนหน่วยประมวลผลกราฟิกส์ได้ ดังรูปที่ 2-11
2. การปรับปรุงการเรียกใช้ส่วนการทำงานให้สามารถทำงานบนหน่วยประมวลผลกราฟิกส์ได้ ดังรูปที่ 2-12

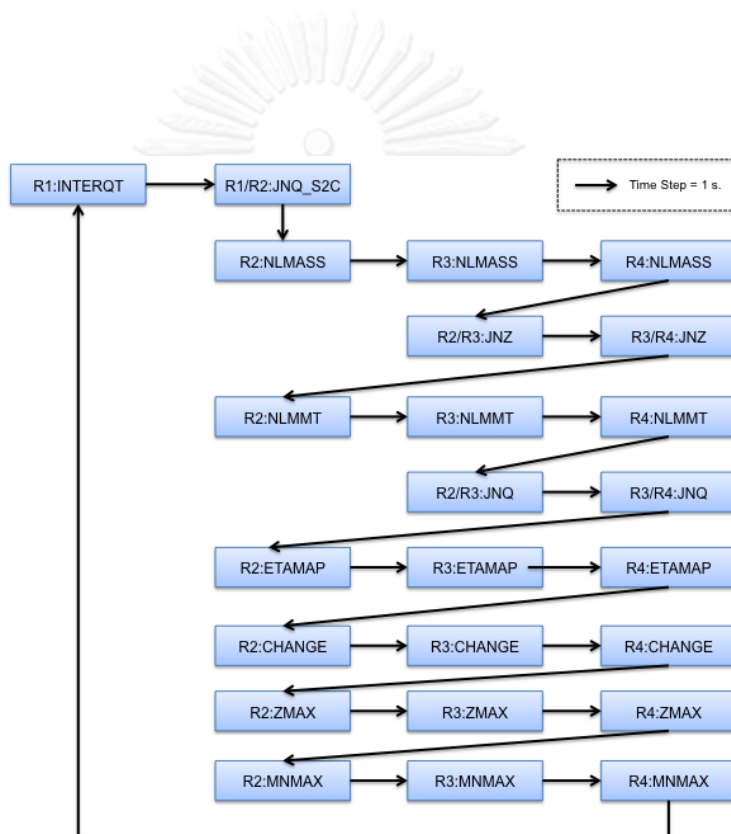
โค้ด Subroutine ZMAX เดิม	โค้ด Subroutine ZMAX ที่แก้ไขเพื่อไปรันบน GPU
	(1) ← MODULE CUDADEV
	(2) ← USE CUDAFOR
	CONTAINS ...
SUBROUTINE ZMAX(IFN,JFN,Z,ZM) C	(3) ← ATTRIBUTES(GLOBAL) SUBROUTINE ZMAXDEV(IFN,JFN,Z,ZM) C
INTEGER IFN, JFN	INTEGER,VALUE :: IFN, JFN
REAL Z(IFN,JFN,2), ZM(IFN,JFN)	REAL,DEVICE :: Z(IFN,JFN,2), ZM(IFN,JFN)
DO J=1, JFN DO I=1, IFN	(4) ← INTEGER :: I,J I = (blockidx%x - 1) * 16 + threadidx%x J = (blockidx%y - 1) * 16 + threadidx%y C
IF (ZM(I,J) .LT. Z(I,J,2)) ZM(I,J) = Z(I,J,2) END DO END DO RETURN	IF (I .LE. IFN .AND. J .LE. JFN) THEN IF (ZM(I,J) .LT. Z(I,J,2)) ZM(I,J) = Z(I,J,2) END IF RETURN

รูปที่ 2-11 ตัวอย่างการปรับปรุงส่วนการทำงาน ZMAX ให้ทำงานด้วย CUDA Fortran



รูปที่ 2-12 ตัวอย่างการเรียกใช้ส่วนการทำงาน ZMAX ให้ทำงานบนหน่วยประมวลผลกราฟิกส์

ซึ่งการทำงานของโปรแกรมจำลองสึนามิในโครงการนี้ไม่ได้ทำงานบนหน่วยประมวลผลกราฟิกส์ทั้งหมด โดยจะทำการคำนวณเฉพาะส่วนมีการคำนวณต่อเนื่องกัน ดังรูปที่ 2-13 และทำการคัดลอกผลลัพธ์ออกจากหน่วยความจำบนหน่วยประมวลผลกราฟิกส์ทุก 60 Time step หรือทุก 60 วินาที(1 นาที) โดยจะเขียนค่า Z, M, N ของทั้ง 4 ระดับความละเอียด ซึ่งเป็นค่าความสูงจากผิวน้ำ ค่าอัตราการไหลของน้ำในแนวแกน x และ y ตามลำดับ และเมื่อสิ้นสุดการคำนวณจะทำการเขียนผลลัพธ์สูงสุด (max) ของค่าต่างๆ และเวลาที่คลื่นจะมาถึงพิกัดนั้นๆออกมาดังเช่นโปรแกรมจำลองสึนามิต้นฉบับ



รูปที่ 2-13 ลำดับการทำงานของโปรแกรมที่คำนวณบนหน่วยประมวลผลกราฟิกส์

ซึ่งผลของการพัฒนาโครงการทางวิศวกรรมดังกล่าวได้แสดงให้เห็นถึงศักยภาพของหน่วยประมวลผลกราฟิกส์ที่สามารถประมวลผลงานทั่วไปได้ดี ซึ่งมีความสามารถในการคำนวณสูงกว่าหน่วยประมวลผลกลางในหนึ่งแกน แต่จะมีความล่าช้าในขั้นตอนการคัดลอกข้อมูลเข้า-ออกหน่วยความจำบนหน่วยประมวลผลกราฟิกส์ ดังนั้นจึงมีความน่าสนใจที่จะพัฒนาโครงสร้างของการประมวลผลที่ผนวกทั้งหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์ให้ร่วมกันทำงานแบบขนานได้อย่างมีประสิทธิภาพ และควรมีวิธีที่จะสามารถจัดงานแบบขนานที่จะสามารถกระจายงานไปยังหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์ได้อย่างเหมาะสม

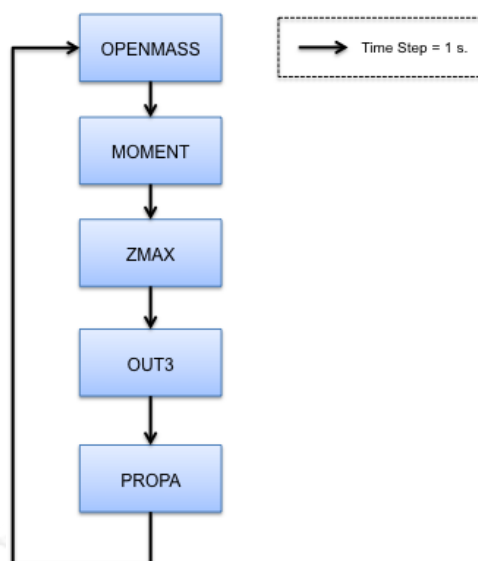


### 2.2.3 Parallelizing TUNAMI-N1 using GPGPU

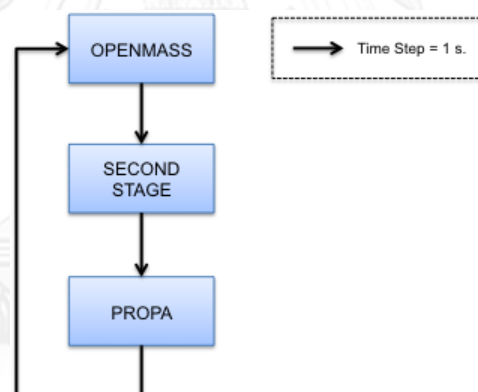
การพัฒนาโปรแกรมจำลองสึนามิ TUNAMI เวอร์ชัน N1 ให้สามารถคำนวณแบบขนานบนหน่วยประมวลผลกราฟิกส์ได้ เป็นผลงานวิจัยของ Harsh Gidra, Israrul Haque, Nitin P Kumar, Sargurunathan M, M. S. Gaur, Vijay Laxmi, M.Zwolinski และ Virendra Singh กลุ่มนักวิจัยภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันวิจัยเทคโนโลยีแห่งชาติมาลาวียา (Malaviya National Institute of Technology) ประเทศอินเดีย ในปี 2554 [18] โดยปัญหาของงานวิจัยนี้เกิดขึ้นจากการที่โปรแกรมจำลองสึนามิในเวอร์ชันเอ็นหนึ่ง ที่เป็นการจำลองระยะใกล้ และมีสมการคำนวณที่เป็นเชิงเส้นโดยเป็นสมการคำนวณสำหรับน้ำตื้น ซึ่งสมการสำหรับน้ำตื้นจะพิจารณาถึงปัจจัยของแรงที่กระทำที่ส่งผลกระทบต่อความสูงของคลื่น โดยในโปรแกรมจำลองสึนามินั้นใช้ข้อมูลปริมาณมากในการคำนวณดังนั้นจึงเหมาะสมที่จะนำมาปรับปรุงให้มีการทำงานแบบขนานเกิดขึ้น

โดยพวกเขาจัดทำการพัฒนาโปรแกรมจำลองสึนามิให้สามารถคำนวณแบบขนานได้โดยถ่ายโอนงานให้หน่วยประมวลผลกราฟิกส์ทำงานผ่านชุดคำสั่งของ CUDA ซึ่งจะประกอบไปด้วยสองแบบจำลองที่พวกเขาได้นำเสนอสำหรับการพัฒนางานแบบขนาน ได้แก่

1. Data-Parallel model : จะทำการคำนวณความสูงของคลื่นในแต่ละจุดด้วยรูปแบบการคำนวณขนาน ซึ่งส่วนการคำนวณพื้นฐานจะถูกคำนวณบนหน่วยประมวลผลกราฟิกส์ ดังรูปที่ 2-14
2. Hybrid model : จะแบ่งบล็อกการทำงานออกเป็นกลุ่มสำหรับประมวลงานที่มีความแตกต่างกันออกไป คือในส่วนการทำงานที่มีชื่อว่า “SECOND STAGE” ภายในจะประกอบไปด้วย 3 ส่วนการทำงานอันได้แก่ 2714 บล็อกแรกจะถูกใช้สำหรับการอนุรักษ์โมเมนตัมบนเมทริกซ์ทั้งหมด (MOMENT) ส่วน 2714 บล็อกถัดมาจะถูกใช้สำหรับการคำนวณรูปแบบการเดินทางของคลื่น (PROPA) และบล็อกที่เหลือจะถูกใช้สำหรับการหาค่าสูงสุดของคลื่นในแต่ละจุด (ZMAX) ดังปรากฏตามรูปที่ 2-15



รูปที่ 2-14 ลำดับการทำงานของแบบจำลองไดนามิก Data-parallel ที่คำนวณบนหน่วยประมวลผลกราฟิกส์



รูปที่ 2-15 ลำดับการทำงานของแบบจำลองไดนามิก Hybrid ที่คำนวณบนหน่วยประมวลผลกราฟิกส์

นอกจากที่จะเสนอแบบจำลองสำหรับการทำงานรูปแบบขนานบนหน่วยประมวลผลกราฟิกส์แล้วนั้น ยังได้กล่าวเพิ่มเติมในเรื่องของเทคนิคการปรับปรุงประสิทธิภาพให้กับโปรแกรมแบบขนานที่ใช้งานชุดคำสั่ง CUDA ตัวอย่างเช่น จำนวนของเทรดที่ทำงานภายในบล็อกควรจะเป็นตัวคูณของ Warp size, การกำหนดจำนวนรีจิสเตอร์ที่ใช้ในการทำงานภายในเทรดเพื่อทำให้จำนวนของเทรดที่สามารถทำงานได้เพิ่มมากขึ้น, การจัดการการเข้าถึงหน่วยความจำโกลบอลในเคอเนล (Global memory) หรือการเข้าถึงหน่วยความจำร่วม (Shared memory), การใช้ตัวเลือก “use-fast-

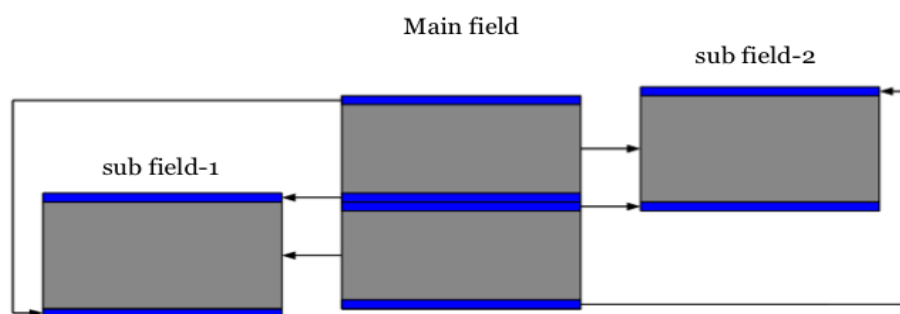
math” ของ fast mathematical library เพื่อเพิ่มความเร็วให้กับการคำนวณแต่จะส่งผลให้ความละเอียดหรือแม่นยำของข้อมูลลดลง

ซึ่งในงานวิจัยดังกล่าวได้ทำการพัฒนาและปรับปรุงให้ผลลัพธ์อยู่ในระดับที่น่าพึงพอใจ และจากการรายงานผลของงานวิจัยจากกลุ่มผู้พัฒนาก็ชี้ให้เห็นถึงความสามารถของการทำงานบนหน่วยประมวลผลกราฟิกส์จะทำงานได้อย่างมีประสิทธิภาพ เมื่อขนาดของกริดนั้นใหญ่มากเพียงพอ และจากเทคนิคที่กลุ่มผู้พัฒนาได้นำเสนอสามารถนำไปประยุกต์ใช้กับการทำงานแบบขนานบนหน่วยประมวลผลกราฟิกส์เพื่อเพิ่มประสิทธิภาพให้กับการคำนวณ อีกทั้งยังช่วยให้ผู้พัฒนาเข้าใจถึงโครงสร้างของสถาปัตยกรรมที่กำลังพัฒนาอยู่ได้เป็นอย่างดี

#### 2.2.4 Asynchronous Communication Schemes for Finite Difference Methods on Multiple GPUs

การสื่อสารแบบไม่ประสานจังหวะสำหรับการจำลองด้วยวิธีผลต่างอันดับ (Finite-Difference) บนกลุ่มของหน่วยประมวลผลกราฟิกส์ เป็นผลงานวิจัยของ D.P. Playne และ K.A. Hawick คณะวิทยาการคอมพิวเตอร์ มหาวิทยาลัยมาสเซย์ (Massey University) ประเทศนิวซีแลนด์ ในปี 2553 [19] โดยปัญหาของงานวิจัยนี้เกิดจากการที่ปัญหาทางวิศวกรรมที่ถูกแก้ไขด้วยวิธีการแบบ Finite-Difference เมื่อถูกนำไปคำนวณบนหน่วยประมวลผลกราฟิกส์ (GPU) ปัญหาที่เกิดขึ้นก็คือความล่าช้าที่เกิดจากการสื่อสารหรือแลกเปลี่ยนข้อมูลระหว่าง GPU กับ CPU เนื่องด้วยจากข้อจำกัดทางอุปกรณ์ ดังนั้นจึงควรจะมีการปรับเปลี่ยนอัลกอริทึมที่ทำการหาคำนวณที่จะทำให้เกิดการสื่อสารที่เหลื่อมกัน (Overlap) ระหว่างการคำนวณ เพื่อเพิ่มประสิทธิภาพให้กับระบบ

และสิ่งที่ปรับปรุงและแก้ไขในงานวิจัยนอกจากอัลกอริทึมก็คือการที่จะแบ่งพื้นที่การจำลองออกเป็นส่วนย่อยๆ เพื่อให้หน่วยประมวลผลกราฟิกส์ทำงาน ตัวอย่างได้แก่การแบ่งพื้นที่ที่ปรากฏดังรูปที่ 2-16 ซึ่งจะทำให้การแบ่งพื้นที่ออกเป็นสองส่วนจากพื้นที่หลักซึ่งอยู่ตรงกลางของภาพ (Main field) ส่วนทางด้านซ้ายและขวา (sub field-1 และ sub field-2) จะเป็นพื้นที่ที่ได้รับการแบ่งออกมาแล้วจะเห็นว่าที่ขอบของข้อมูลจะมีเซลล์ขอบของพื้นที่ที่ติดกันนั้นรวมอยู่ด้วย



รูปที่ 2-16 ตัวอย่างการแบ่งพื้นที่ออกจากพื้นที่หลักเพื่อใช้งานบนหน่วยประมวลผลกราฟิกส์

ส่วนในเรื่องของอัลกอริทึมที่ใช้ในงานวิจัยก็จะแสดงให้เห็นทางด้านขวาของตารางที่ 2-2 โดยจะปรับเปลี่ยนจากอัลกอริทึมแบบทั่วไป ซึ่งจะทำการคำนวณทุกเซลล์ที่ได้รับมาจากการแบ่งงาน ให้แบ่งการทำงานออกเป็นสองส่วนคือทำการคำนวณส่วนขอบของข้อมูลไปพร้อมกับคำนวณข้อมูลส่วนที่เหลือเพื่อให้ไม่เป็นการเสียเวลาที่จะต้องรอให้การคำนวณทั้งหมดเสร็จสิ้นถึงจะสามารถแลกเปลี่ยนข้อมูลได้

ตารางที่ 2-2 ตารางเปรียบเทียบอัลกอริทึมในการทำงานระหว่างการคำนวณพื้นที่บนหน่วยประมวลผลกราฟิกส์

Synchronous Memory Copy	Asynchronous Memory Copy
<ul style="list-style-type: none"> <li>• คำนวณแบบจำลองในทุกๆเซลล์</li> <li>• ซิงโครไนซ์กับเทรตอื่นๆ</li> <li>• คัดลอกข้อมูลขอบจาก GPU</li> <li>• แลกเปลี่ยนข้อมูลของเซลล์ขอบ</li> <li>• โหลดข้อมูลขอบใหม่ไปยัง GPU</li> <li>• ทำซ้ำ</li> </ul>	<ul style="list-style-type: none"> <li>• คำนวณเซลล์ที่อยู่ขอบของฟิลด์ย่อย (Stream 1)</li> <li>• คำนวณเซลล์ที่เหลือในฟิลด์ (Stream 2)</li> <li>• คัดลอกข้อมูลขอบไปยังโฮสต์ (Stream 1)</li> <li>• แลกเปลี่ยนข้อมูลของเซลล์ขอบ (CPU)</li> <li>• โหลดข้อมูลขอบใหม่ไปยัง GPU (Stream 1)</li> <li>• ทำซ้ำ</li> </ul>

ผลจากการวิจัยที่ทำให้ได้ทราบถึงอัลกอริทึมดังกล่าวที่สามารถช่วยลดเวลาในการทำงานลดได้อันเนื่องมาจากการแยกกันทำงานของส่วนที่สามารถแยกกันทำงานได้บนหน่วยประมวลผลกราฟิกส์ ซึ่งการแยกกันของการทำงานไม่ได้เป็นการให้งานนั้นทำคนละเวลาแต่งานที่แยกออกมาสามารถประมวลผลได้อย่างพร้อมกัน จึงเป็นประโยชน์อย่างยิ่งสำหรับกรณีที่นำไปประยุกต์ใช้กับการทำงานบนหน่วยประมวลผลกราฟิกส์ ซึ่งจะสามารถทำให้ความเร็วในการคำนวณลดลงและเพิ่มประสิทธิภาพให้กับระบบการทำงานได้เป็นอย่างดี โดยผู้พัฒนาต้องมีความเข้าใจถึงโครงสร้างของข้อมูลและโปรแกรมว่าจะสามารถแยกส่วนการทำงานออกมาได้มากน้อยแค่ไหนถึงจะก่อให้เกิดประโยชน์สูงสุด

## บทที่ 3

### กรณีศึกษา

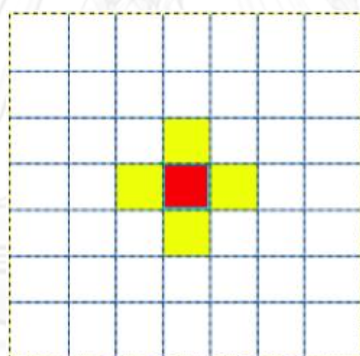
#### 3.1 กรณีศึกษาที่ใช้ในงานวิจัย

สำหรับการออกแบบและพัฒนาวิธีจัดการงานของโปรแกรมคำนวณแบบขนานสำหรับปัญหาที่มีหลายระดับความละเอียดของงานวิจัยนี้ได้เลือกใช้การจำลองสึนามิที่มีชื่อว่า “TUNAMI” เป็นกรณีศึกษา ซึ่งโปรแกรมดังกล่าวแต่เดิมเป็นโปรแกรมคำนวณแบบลำดับและได้รับการพัฒนาให้อยู่ในรูปแบบขนานเพื่อบรรลุวัตถุประสงค์ในการลดเวลาในการทำงานของโปรแกรม โดยจะอธิบายลักษณะการทำงานของโปรแกรม ลักษณะของข้อมูลที่ใช้ผ่าน 2 งานวิจัยดังนี้

##### 3.1.1 โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับ (TUNAMI Program)

โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับ หรือโปรแกรม TUNAMI (Tohoku University's Numerical Analysis Model for Investigation of tsunamis) [20] เป็นโปรแกรมจำลองสึนามิที่พัฒนาด้วยชุดคำสั่งภาษาฟอร์แทรน (FORTRAN) และเป็นโปรแกรมที่มีการคำนวณแบบลำดับ (Sequential Program) ซึ่งเป็นผลงานวิจัยของ Dr. Fuhimiko Imamura จากมหาวิทยาลัยโตโฮกุ (Tohoku University) ประเทศญี่ปุ่น กับ Dr. Armet Cevdet Yalciner และผู้ช่วยนักวิจัย Gulizar Ozyurt จากมหาวิทยาลัยมิดเดิลอีสต์เทคนิคัล ประเทศตุรกี และผศ.ดร.อาณัติ เรืองรัมย์ ศูนย์เชี่ยวชาญเฉพาะทางด้านวิศวกรรมแผ่นดินไหวและการสั่นสะเทือน ภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย เป็นผู้นำมาปรับปรุงและพัฒนาต่อเพื่อนำมาใช้ในการเตือนภัยสึนามิจากเหตุการณ์แผ่นดินไหวในเขตมหาสมุทรอินเดีย โดยความถูกต้องแม่นยำจากผลลัพธ์ของโปรแกรมเมื่อนำมาใช้งานเป็นที่น่าพอใจ แต่โปรแกรมจำลองสึนามินี้จะใช้เวลาในการจำลองสึนามิในแต่ละกรณีประมาณ 4-6 ชั่วโมง ทั้งนี้จะขึ้นกับขนาดของแต่ละระดับความละเอียดของพื้นที่ที่ใช้การคำนวณและสมรรถนะของเครื่องคอมพิวเตอร์ที่ใช้ โดยการนำโปรแกรมจำลองสึนามิไปใช้งานมีด้วยกัน 2 วิธี ได้แก่ วิธีแรกนำโปรแกรมไปคำนวณกรณีที่อาจจะเกิดขึ้นได้จำนวนมาก เมื่อเกิดแผ่นดินไหวใกล้เคียงกับจุดใดที่เคยจำลองไว้ก็จะนำผลลัพธ์ที่เก็บไว้ในฐานข้อมูลมาใช้ประกอบการตัดสินใจเตือนภัยซึ่งกรณีที่ต้องทำการจำลองเพื่อเก็บผลลัพธ์ไว้มีจำนวนมากกว่า 1,000 กรณี และวิธีที่สองนำโปรแกรมไปคำนวณในทันทีที่เกิดแผ่นดินไหว เพื่อประกอบการตัดสินใจเตือนภัยซึ่งมีความแม่นยำกว่าการใช้ผลลัพธ์จากการคำนวณล่วงหน้า ซึ่งในวิธีที่สองนี้จำเป็นต้องคำนวณอย่างรวดเร็วเพื่อตอบสนองข้อมูลก่อนที่สึนามิจะเกิดขึ้นจริง

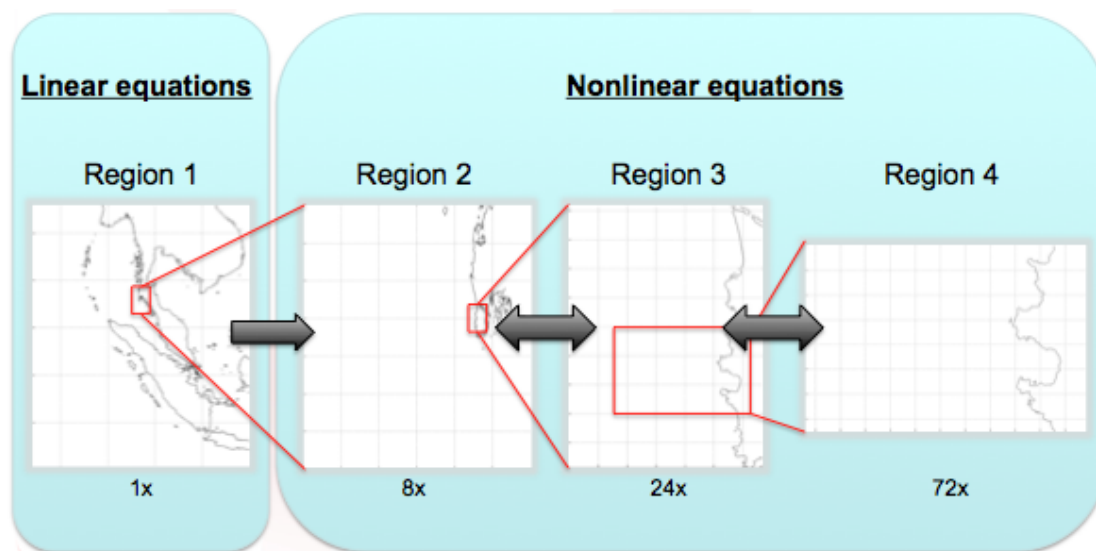
ในการคำนวณของโปรแกรมจำลองสึนามิ ได้แบ่งการคำนวณออกเป็นสองส่วน คือ การจำลองคลื่นสึนามิระยะไกลในระบบพิกัดทรงกลม (Far-field Tsunami Simulation in Spherical Coordinate System) และการจำลองคลื่นสึนามิระยะใกล้ในระบบพิกัดคาร์ทีเซียน (Near-field Tsunami Simulation in Cartesian Coordinate System) ด้วยการคำนวณแบบวิธีการวิเคราะห์เชิงตัวเลขแบบ Finite Difference Method ที่ต้องอาศัยข้อมูลของจุดจากบริเวณรอบๆจุดที่กำลังสนใจมาร่วมในการคำนวณด้วยเพื่อที่จะทราบถึงความเปลี่ยนแปลงของค่า ณ จุดที่กำลังสนใจอยู่ ดังรูปที่ 3-1 และเนื่องจากในแต่ละบริเวณมีความต้องการความละเอียดของข้อมูลที่แตกต่างกัน เช่น พื้นที่ชายฝั่งต้องการความละเอียดของข้อมูลมากกว่าบริเวณกลางทะเล ดังนั้นเมื่อความละเอียดของข้อมูลสูงขึ้นเวลาที่ใช้ในการคำนวณก็จะมากขึ้นตามไปด้วย ในโปรแกรม TUNAMI ได้ใช้วิธีการคำนวณแบบหลายระดับความละเอียด (Multi-scale) ซึ่งมีการแลกเปลี่ยนข้อมูลกันระหว่างการคำนวณในแต่ละระดับความละเอียดบริเวณรอยต่อของพื้นที่ เพื่อให้เกิดความต่อเนื่องกันของข้อมูลในแต่ละระดับความละเอียด



รูปที่ 3-1 ตัวอย่างรูปแบบการวิเคราะห์เชิงตัวเลขแบบ Finite Difference Method

โดยในแต่ละระดับความละเอียดจะเรียกว่า Region ซึ่งจะใช้สัญลักษณ์ R1, R2, R3 และ R4 แทนแต่ละระดับความละเอียด ดังรูปที่ 3-2 แสดงตัวอย่างของข้อมูลบริเวณที่นำมาคำนวณที่มี 4 ระดับความละเอียด ผลของการจำลองของแต่ละระดับความละเอียดก็ขึ้นกับลักษณะของการเตือนภัย โดยพื้นที่ที่มีความละเอียดสูงที่สุดเมื่อทำการคำนวณการท่วมขึ้นบนชายฝั่งของน้ำทะเล (Run-up) ผลลัพธ์ที่ได้จะสามารถบอกถึงความเสียหายที่จะเกิดขึ้นได้อย่างแม่นยำ และจะเรียกพื้นที่ที่มีความละเอียดสูงที่สุด หรือพื้นที่ในระดับ R4 ว่า “โซน” โดยโปรแกรมจำลองสึนามิตัวนี้จะเจาะจงพื้นที่ในระดับ R4 เพียงจุดเดียว ที่มีพื้นที่ในระดับ R3, R2 และ R1 ครอบคลุมอยู่ เมื่อต้องการเปลี่ยนพื้นที่ใน

ระดับ R4 ไปยังตำแหน่งอื่น โปรแกรมจำลองสึนามิก็ต้องคำนวณพื้นที่ที่ครอบคลุมใหม่ตามการเปลี่ยนแปลง



รูปที่ 3-2 รายละเอียดของพื้นที่ที่ใช้งานในโปรแกรม TUNAMI

เนื่องจากโปรแกรม TUNAMI สามารถทำงานได้เพียงที่ละโซนดังรูปที่ 3-2 ดังนั้นหากต้องทำการจำลองสึนามิจากการเกิดแผ่นดินไหวใต้ทะเลให้ครอบคลุมทั้งพื้นที่ดังตารางที่ 3-1 จำเป็นต้องทำการคำนวณพื้นที่ทั้งหมด 18 โซน ดังนี้

1. โซน 2111 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 211 > 2111$
2. โซน 2112 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 211 > 2112$
3. โซน 2121 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 212 > 2121$
4. โซน 2122 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 212 > 2122$
5. โซน 2131 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 213 > 2131$
6. โซน 2141 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 214 > 2141$
7. โซน 2151 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 215 > 2151$
8. โซน 2152 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 215 > 2152$
9. โซน 2153 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 21 > 215 > 2153$
10. โซน 2211 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 22 > 221 > 2211$
11. โซน 2212 ใช้ข้อมูลในระดับความละเอียดหมายเลข  $1 > 22 > 221 > 2212$

12. โชน 2213 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 22 > 221 > 2213
13. โชน 2311 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 231 > 2311
14. โชน 2312 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 231 > 2312
15. โชน 2313 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 231 > 2313
16. โชน 2321 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 232 > 2321
17. โชน 2322 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 232 > 2322
18. โชน 2323 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 232 > 2323

ตารางที่ 3-1 พิกัดของบริเวณที่นำมาคำนวณด้วยโปรแกรมจำลองสึนามิ

ระดับความละเอียด	หมายเลข	ความละเอียด		ขอบเขต												ขนาดข้อมูลตามแกน		ตำแหน่งบนระดับความละเอียดที่น้อยกว่า					
				ละติจูด						ลองจิจูด													
				จาก			ถึง			จาก			ถึง					จาก		ถึง			
				'	"	°	'	"	°	'	"	°	'	"	°			'	"	X	Y	X	Y
R1	1	2	0	-10	0	0	18	0	0	87	0	0	110	0	0	690	840	-	-	-	-		
R2	21	0	15	5	59	45	9	30	0	95	59	45	99	0	0	721	841	271	481	360	585		
	22	0	15	9	31	45	13	2	0	95	59	45	99	0	0	721	841	271	587	360	691		
	23	0	15	4	45	45	8	16	0	99	1	45	102	2	0	721	841	362	444	451	548		
R3	211	0	5	7	45	25	8	12	30	97	59	55	98	20	0	241	325	482	424	561	531		
	212	0	5	8	32	25	9	11	30	97	26	55	98	20	0	637	469	350	612	561	767		
	213	0	5	9	11	55	9	29	30	97	58	55	98	32	0	397	211	478	770	609	839		
	214	0	5	8	12	55	8	32	0	97	59	55	98	20	0	241	229	482	534	561	609		
	215	0	5	7	31	55	8	30	0	98	20	25	98	59	30	469	697	564	370	719	601		
	221	0	5	9	32	55	10	13	30	97	47	0	98	40	15	639	487	430	6	642	167		
	231	0	5	7	7	25	7	46	15	99	3	25	99	58	0	655	466	8	568	225	722		
	232	0	5	6	27	55	7	6	45	99	8	55	100	2	0	637	466	30	410	241	564		
R4	2111	0	1.667	7	50	58	8	0	0	98	5	28	98	19	0	487	325	68	68	229	175		
	2112	0	1.667	8	0	18	8	12	0	98	5	28	98	19	0	487	421	68	180	229	319		
	2121	0	1.667	8	36	58	8	50	0	98	0	8	98	18	0	643	469	400	56	613	211		



	2122	0	1.667	8 50 18	9 11 0	98 0 8	98 19 0	679	745	400	216	625	463
	2131	0	1.667	9 12 18	9 29 0	98 11 28	98 28 0	595	601	152	6	349	205
	2141	0	1.667	8 13 18	8 31 0	98 8 28	98 19 0	379	637	104	6	229	217
	2151	0	1.667	7 47 18	8 10 0	98 21 28	98 29 0	271	817	14	186	103	457
	2152	0	1.667	8 10 18	8 24 0	98 26 28	98 47 0	739	493	74	462	319	625
	2153	0	1.667	7 55 18	8 10 0	98 42 28	98 57 0	523	529	266	282	439	457
	2211	0	1.667	10 0 18	10 13 0	98 20 28	98 40 0	703	457	403	330	636	481
	2212	0	1.667	9 46 18	10 0 0	98 20 28	98 40 0	703	493	403	162	636	325
	2213	0	1.667	9 34 18	9 46 0	98 20 28	98 40 0	703	421	403	18	636	157
	2311	0	1.667	7 33 18	7 46 0	99 4 28	99 23 0	667	457	14	312	235	463
	2312	0	1.667	7 20 18	7 33 0	99 10 28	99 28 0	631	457	86	156	295	307
	2313	0	1.667	7 8 18	7 20 0	99 25 28	99 45 0	703	421	266	12	499	151
	2321	0	1.667	6 53 18	7 6 0	99 25 28	99 45 0	703	457	200	306	433	457
	2322	0	1.667	6 43 18	6 53 0	99 35 28	99 53 0	631	349	320	186	529	301
	2323	0	1.667	6 29 18	6 43 0	99 43 28	100 1 0	631	493	416	18	625	181

ข้อมูลที่น่ามาใช้ในการจำลองสึนามิได้ทำการเก็บและสร้างข้อมูลจำลองสำหรับใช้ในการคำนวณจากข้อมูลที่สำรวจขึ้นจริง โดยมี ผศ.ดร.อาณัติ เรืองรัมย์ และนิสิตภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัยเป็นผู้จัดทำขึ้นมาเป็นแฟ้มข้อมูล ซึ่งแฟ้มข้อมูลระดับความสูงของคลื่นผิวน้ำจะเรียกว่า Deform และแฟ้มข้อมูลความลึกของน้ำทะเลจะเรียกว่า Region พร้อมทั้งระบุพิกัดและขนาดของพื้นที่ในแต่ละบริเวณ และจะระบุหมายเลขที่จะแสดงถึงระดับความละเอียดตามตารางที่ 3-1 ที่แสดงพิกัดทั้งหมด 30 พื้นที่ซึ่งเป็นออกเป็น 4 ระดับความละเอียด พื้นที่ในระดับ R4 มีจำนวน 18 พื้นที่ ถูกครอบคลุมด้วยพื้นที่ในระดับ R3 จำนวน 8 พื้นที่ และพื้นที่ในระดับ R2 จำนวน 3 พื้นที่ โดยทุกพื้นที่จะถูกครอบคลุมด้วยพื้นที่ในระดับ R1 ซึ่งระดับของความละเอียดจะเป็นไปดังลักษณะนี้ พื้นที่ในระดับ R1 ไป R2 เพิ่มขึ้น 8 เท่า พื้นที่ในระดับ R2 ไป R3 และพื้นที่ในระดับ R3 ไป R4 เพิ่มขึ้น 3 เท่า ดังปรากฏในรูปที่ 3-2

การทำงานของโปรแกรมการจำลองนั้นการคำนวณในแต่ละรอบจะเรียกว่า Time-Step ซึ่งแทนการจำลองเหตุการณ์ใน 1 วินาที โดยการคำนวณของพื้นที่ในระดับ R1 จะคำนวณทุกๆ 4 วินาที

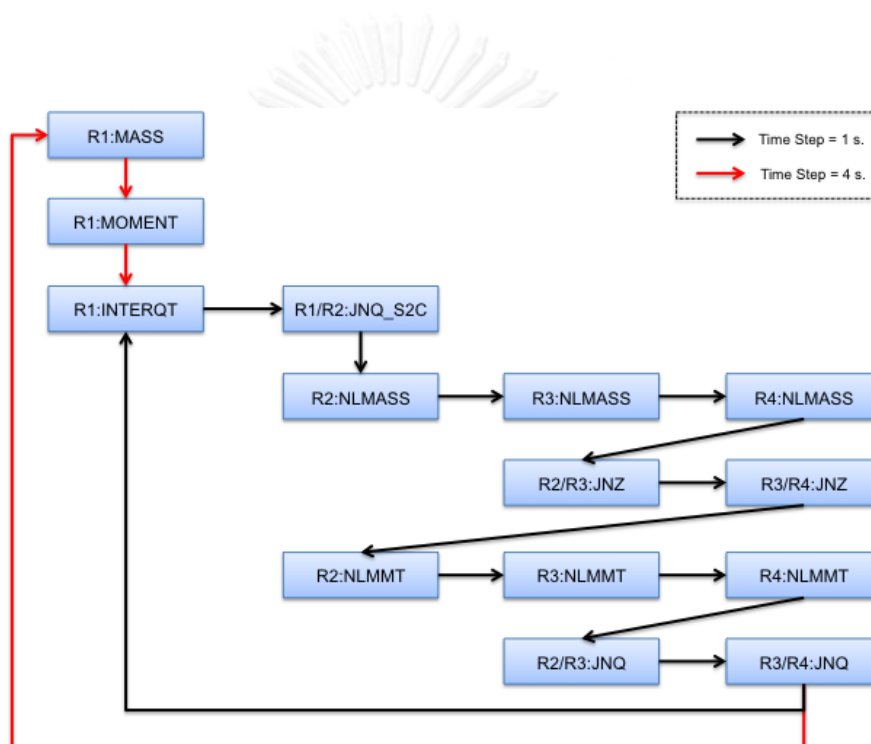
และพื้นที่ในระดับ R2, R3 และ R4 จะคำนวณทุกๆ 1 วินาที ซึ่งจะทำการจำลองทั้งหมด 36,000 วินาที หรือ 10 ชั่วโมงนับจากเวลาที่เริ่มเกิดแผ่นดินไหวได้ทะเล จากนั้นจะทำการบันทึกข้อมูลเพื่อใช้ในการสร้างภาพทุกๆ 60 วินาทีหรือ 1 นาที ซึ่งค่าดังกล่าวสามารถทำการแก้ไขได้จากแฟ้มข้อมูลขาเข้าของโปรแกรม โดยค่าเวลาต่างๆที่กำหนดเป็นค่าเวลาที่ทางภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย ได้กำหนดขึ้นมาไว้ตามความเหมาะสมในการจำลองสึนามิเพื่อการเตือนภัย แต่เพื่อประสิทธิภาพในการเตือนภัยที่สูงขึ้น เวลาที่ใช้ในการคำนวณควรจะลดลงเพื่อให้สามารถทราบถึงจุดที่อาจจะได้รับผลกระทบจากคลื่นสึนามิและเตือนภัยให้อพยพก่อนที่จะเกิดเหตุการณ์ที่คาดไม่ถึง ดังนั้นจึงจำเป็นที่จะต้องหาวิธีการเพื่อให้โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับขั้นนี้ใช้เวลาลดลง

### 3.1.2 โปรแกรมแบบขนานเพื่อจำลองการเกิดสึนามิ (Parallel Tsunami Simulation Program)

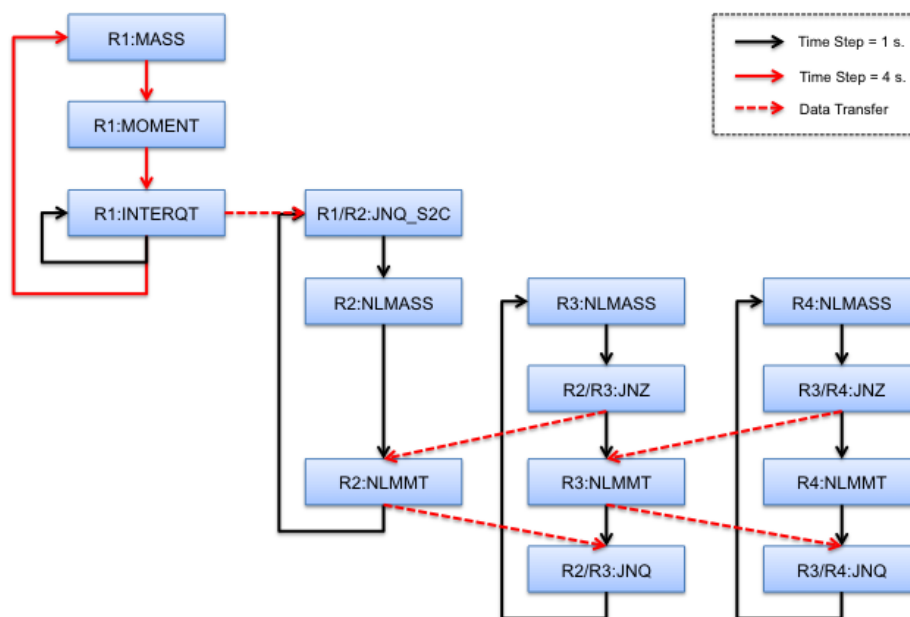
โปรแกรมแบบขนานเพื่อจำลองการเกิดสึนามินี้ได้รับการพัฒนาในโครงการทางวิศวกรรมของนายกิตติพัฒน์ วิโรจน์ศิริ นิสิตภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งเป็นผู้พัฒนา และผศ.ดร.วีระ เหมืองสิน เป็นอาจารย์ที่ปรึกษาโครงการ ในปี 2550 [21] โดยปัญหาของโครงการนี้เกิดขึ้นมาจากการที่โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับขั้นใช้เวลาในการคำนวณประมาณ 4-6 ชั่วโมงในแต่ละโซน ถ้าต้องคำนวณการเกิดสึนามิพื้นที่ทั้งหมด 18 โซน ด้วยกรณีของการเกิดแผ่นดินไหวได้ทะเลมากกว่า 1,000 กรณี ก็จะใช้เวลามากขึ้นอีก 1,000 เท่าและผลลัพธ์จากการคำนวณในแต่ละกรณีก็ซ้ำซ้อนกัน ดังนั้นจึงมีการพัฒนาโปรแกรมจำลองสึนามิให้พัฒนาการคำนวณอยู่ในรูปแบบขนานที่จะสามารถลดเวลาในการคำนวณเพื่อเพิ่มประสิทธิภาพให้กับโปรแกรมจำลองสึนามิ ซึ่งในโครงการจะปรับปรุงรูปแบบการคำนวณโดยใช้ Message Passing Interface (MPI) เป็นเครื่องมือในการคำนวณแบบขนานบนระบบคลัสเตอร์คอมพิวเตอร์ ซึ่งโปรแกรมจำลองสึนามิที่ได้รับการพัฒนาได้รับการทดสอบบนระบบคลัสเตอร์ TERA ของ Thai National Grid Center มหาวิทยาลัยเกษตรศาสตร์ ประเทศไทย และระบบคลัสเตอร์ TSUBAME ของ GSIC Center สถาบันเทคโนโลยีแห่งโตเกียว (Tokyo Institute of Technology / TiTech) ประเทศญี่ปุ่น โปรแกรมจำลองสึนามิที่คำนวณแบบขนานนี้จะทำการคำนวณด้วยฟังก์ชันเดิมจากโปรแกรมจำลองสึนามิที่คำนวณแบบลำดับขั้นเขียนด้วยชุดคำสั่งภาษาฟอร์แทรน แต่จะมีการปรับปรุงในส่วนของการเรียกใช้และควบคุมการทำงานแบบขนานด้วย MPI ซึ่งจะเขียนด้วยชุดคำสั่งภาษาซีพลัสพลัส (C++)

แนวทางในการแก้ปัญหาของโครงการนี้จะแบ่งออกเป็นสองขั้นตอนหลัก ซึ่งขั้นตอนแรกจะแบ่งการคำนวณของพื้นที่ในแต่ละระดับความละเอียดให้แต่ละหน่วยประมวลผลรับผิดชอบการทำงาน ซึ่งจะต้องคำนึงถึงการเชื่อมต่อกันของข้อมูลหรือมีการใช้งานข้อมูลร่วมกันระหว่างระดับความละเอียด

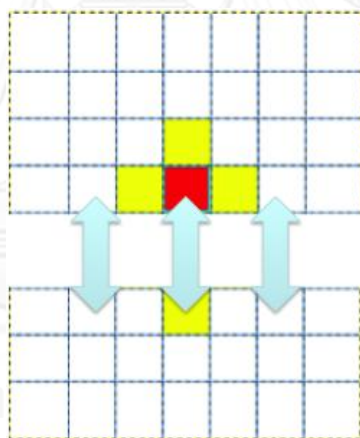
ดังรูปที่ 3-3 ซึ่งจะแสดงถึงลำดับการทำงานของโปรแกรมที่คำนวณแบบลำดับซึ่งมีความเชื่อมต่อกันของข้อมูลระหว่างระดับความละเอียด และรูปที่ 3-4 จะแสดงถึงลำดับการทำงานด้วยการคำนวณแบบขนาน ซึ่งจะแยกกันทำงานออกไปตามหน่วยประมวลผล เส้นประสีแดงจะแสดงถึงการส่งผ่านข้อมูลระหว่างหน่วยประมวลผล ขั้นตอนที่สองจะแบ่งการคำนวณของแต่ละระดับความละเอียดออกเป็นงานย่อยๆ ให้แต่ละหน่วยประมวลผลทำงานดังรูปที่ 3-5 ซึ่งจะสามารถช่วยลดปริมาณงานที่คำนวณต่อหนึ่งหน่วยประมวลผลลงและทำให้ระยะเวลาในการคำนวณของโปรแกรมลดลงด้วย



รูปที่ 3-3 ลำดับการทำงานของโปรแกรมคำนวณแบบลำดับในส่วนที่มีการเชื่อมต่อกันของข้อมูล

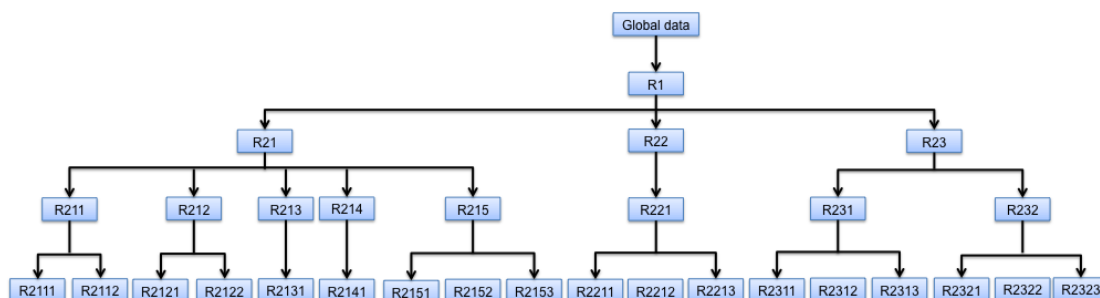


รูปที่ 3-4 ลำดับการทำงานของโปรแกรมด้วยการคำนวณแบบขนาน



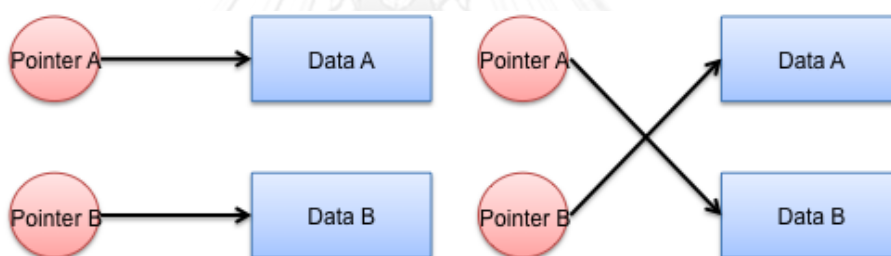
รูปที่ 3-5 ตัวอย่างการแบ่งการคำนวณของแต่ละระดับความละเอียดออกเป็นงานย่อยๆ

ข้อมูลที่เก็บโครงสร้างของพื้นที่ที่จะคำนวณจะจัดเก็บโดยแฟ้มที่ชื่อว่า Config.xml ซึ่งจะมีโครงสร้างแบบต้นไม้ดังรูปที่ 3-6 โดยรากของต้นไม้จะเป็นข้อมูลตัวแปรต่างๆที่ใช้ในการคำนวณทั้งหมด และลูกคือพื้นที่ในระดัความละเอียด R1 ซึ่งจะมีลูกเป็นพื้นที่ในระดัความละเอียด R2, R3 และ R4 ตามลำดับซึ่งจะมีความละเอียดเพิ่มมากขึ้น ซึ่งจะใช้กำหนดพื้นที่ในการคำนวณของแต่ละหน่วยประมวลผล



รูปที่ 3-6 โครงสร้างแบบต้นไม้ของข้อมูลของโปรแกรมจำลองสื่อบนแบบขนาน

นอกจากนี้จะมีการใช้วิธี Page flipping ซึ่งช่วยลดเวลาการทำงานของส่วนโปรแกรมน้อย CHANGE อีกด้วย โดยจะเป็นวิธีการสลับตำแหน่งที่ชี้ข้อมูลของตัวแปรแทนการเขียนทับข้อมูลในตัวแปรทั้งหมดดังรูปที่ 3-7



รูปที่ 3-7 ตัวอย่างวิธีการ Page flipping

ซึ่งในโครงการดังกล่าวได้ทำการพัฒนาและปรับปรุงโปรแกรมจนมีผลลัพธ์ที่น่าพึงพอใจในเรื่องของความเร็วในการคำนวณ แต่ยังคงขาดวิธีการจัดการงานในการคำนวณแบบขนานอย่างมีประสิทธิภาพและความละเอียดของพื้นที่ในแต่ละระดับความละเอียดก็แตกต่างกัน ประสิทธิภาพ รวมทั้งจำนวนรอบที่ทำงานที่แตกต่างกันของการจำลองระยะไกลและระยะใกล้ จึงควรที่ศึกษาเพื่อให้ทราบถึงรูปแบบของแบ่งงานและกระจายงานอย่างเหมาะสมสำหรับหน่วยประมวลผล เพื่อที่จะใช้เป็นวิธีการจัดการงานในการคำนวณแบบขนานที่มีประสิทธิภาพ

## บทที่ 4

### แนวคิดการทำวิจัย

#### 4.1 แนวคิดหลัก

งานวิจัยนี้มีเป้าหมายในการปรับปรุงรูปแบบของข้อมูลที่ใช้ในการคำนวณของโปรแกรมที่คำนวณแบบขนานที่ปัญหาหลายระดับความละเอียดให้เหมาะสมกับความสามารถของระบบคลัสเตอร์คอมพิวเตอร์แบบไฮบริดที่มีทั้งหน่วยประมวลผลแบบมัลติคอร์และหน่วยประมวลผลกราฟิกส์ อีกทั้งยังสามารถจัดสรรงานให้เหมาะสมตามข้อกำหนดในเรื่องของเวลาและหน่วยประมวลผลที่ใช้งานโดยใช้โปรแกรมจำลองสีนามิเป็นกรณีศึกษา และมีพื้นฐานของโปรแกรมมาจากเวอร์ชันที่เป็นการคำนวณแบบขนานที่ใช้หน่วยประมวลผลกลางในการคำนวณ เพื่อที่จะบรรลุเป้าหมายของงานวิจัยจึงต้องทำการพัฒนาโปรแกรมในรูปแบบขนานให้สามารถทำงานบนระบบคลัสเตอร์คอมพิวเตอร์แบบไฮบริดได้ และศึกษาปัจจัยหลักที่ส่งผลต่อประสิทธิภาพของโปรแกรมคำนวณแบบขนานที่ปัญหาหลายระดับความละเอียดที่ทำให้ได้ความเร็วในการคำนวณและมีประสิทธิภาพในการทำงานเพิ่มขึ้น

และแม้ว่างานวิจัยนี้จะเกี่ยวกับคลัสเตอร์ที่แต่ละเครื่องใช้หน่วยประมวลผลแบบ multicore และมี GPU ซึ่งมีวิธีเขียนโปรแกรมที่เป็นไปได้ 3 แบบ คือ message passing (MPI), thread programming (OpenMP) และ data parallel (CUDA) แต่งานวิจัยนี้จะใช้เพียง 2 แบบคือ MPI และ CUDA เพื่อลดความซับซ้อนและ MPI ก็สามารถใช้งานได้ดีกับหน่วยประมวลผลแบบ multicore

โดยงานวิจัยนี้มีแนวคิดในการออกแบบวิธีการจัดการงานแบบขนาน แนวคิดในการทดสอบแนวคิดในการประเมินผล ดังนี้

##### 4.1.1 แนวคิดในการออกแบบวิธีการจัดการงานแบบขนาน

แนวคิดในการออกแบบวิธีการจัดการงานของโปรแกรมที่คำนวณแบบขนาน มีดังนี้

1. พัฒนาโปรแกรมแบบขนานที่ใช้หน่วยประมวลผลแบบมัลติคอร์ในการคำนวณให้สามารถทำงานบนระบบคลัสเตอร์คอมพิวเตอร์แบบไฮบริดได้ โดยสามารถทำการคำนวณได้ทั้งบนหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์ พร้อมทั้งค้นหาสัดส่วนที่เหมาะสมของงานระหว่างหน่วยประมวลผลกราฟิกส์กับหน่วยประมวลผลกลาง

2. วิเคราะห์ประสิทธิภาพของโปรแกรมแบบขนานในด้านต่างๆเพื่อสร้างสมการความสัมพันธ์เวลาของแต่ละส่วนการทำงานในระบบคลัสเตอร์คอมพิวเตอร์นั้นๆ ด้วยการเก็บข้อมูลเวลาของส่วนการทำงานทั้งหมด
3. จัดสรรงานให้เหมาะสมตามข้อกำหนดในเรื่องของเวลาและหน่วยประมวลผลที่ใช้งาน โดยการคำนวณหาค่าปริมาณงานโดยเฉลี่ยของพื้นที่แล้วกำหนดให้แต่ละหน่วยประมวลผล
4. สร้าง timing model หรือสูตรการคำนวณเวลา ซึ่งอยู่ในรูปสมการ เช่น  $T = f(N, P)$  เมื่อ  $N$  คือขนาดของปัญหา และ  $P$  คือจำนวนหน่วยประมวลผล และวัตถุประสงค์คือเมื่อกำหนด  $N$  แล้ว ต้องการหา  $P$  ที่ทำให้  $T$  มีค่าน้อยที่สุด หรือเมื่อกำหนด  $N$  และ  $T$  แล้ว หาค่าของ  $P$  ที่น้อยที่สุด

#### 4.1.2 แนวคิดในการทดสอบ

สำหรับแนวคิดในการทดสอบของงานวิจัยนี้ จะใช้ข้อกำหนดในเรื่องของระยะเวลาในการคำนวณและจำนวนหน่วยประมวลผลที่ใช้ เพื่อให้ทราบถึงประสิทธิภาพเชิงเวลาของแต่ละส่วนการทำงานจากการคำนวณบนระบบคลัสเตอร์คอมพิวเตอร์ เช่น เวลาที่ใช้ในการคำนวณฟังก์ชัน เวลาที่ใช้ในการติดต่อสื่อสารระหว่างหน่วยประมวลผล จากรูปแบบการคำนวณที่แตกต่างกัน ดังนี้

1. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบลำดับที่ใช้การคำนวณหนึ่งโชนโดยใช้หน่วยประมวลผลหนึ่งหน่วย
2. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบขนานที่ใช้การคำนวณหนึ่งโชนและทำการแบ่งงานโดยใช้หน่วยประมวลผลหนึ่งหน่วยต่อหนึ่งพื้นที่
3. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบขนานที่ใช้การคำนวณหนึ่งโชนและทำการแบ่งงานโดยใช้หน่วยประมวลผลมากกว่าหนึ่งหน่วยต่อหนึ่งพื้นที่
4. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบขนานที่ใช้การคำนวณหลายโชนพร้อมกันและทำการแบ่งงานโดยใช้หน่วยประมวลผลหนึ่งหน่วยต่อหนึ่งพื้นที่
5. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบขนานที่ใช้การคำนวณหลายโชนพร้อมกันและทำการแบ่งงานโดยใช้หน่วยประมวลผลมากกว่าหนึ่งหน่วยต่อหนึ่งพื้นที่
6. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบขนานที่ใช้การคำนวณหนึ่งโชนและทำการแบ่งงานโดยใช้หน่วยประมวลผลกราฟิกส์ต่อหนึ่งพื้นที่
7. ทดสอบด้วยโปรแกรมจำลองสีนามิที่คำนวณแบบขนานที่ใช้การคำนวณหนึ่งโชนและทำการแบ่งงานโดยใช้หน่วยประมวลผลกราฟิกส์ต่อหนึ่งส่วนจากการแบ่งพื้นที่

ในการทดสอบที่ 1 จะเป็นการทดสอบการคำนวณที่ไม่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผล เพื่อให้ทราบถึงเวลาที่ใช้ในการคำนวณจริง และเพื่อเป็นฐานในการเปรียบเทียบความเร็วที่เพิ่มขึ้นจากการคำนวณ และในการทดสอบที่ 2 ถึง 5 จะเป็นการทดสอบการคำนวณเพื่อให้ทราบถึงระยะเวลาที่ใช้ในการติดต่อสื่อสารกันระหว่างหน่วยประมวลผล และระยะเวลาที่ใช้ในการคำนวณจริงของแต่ละหน่วยประมวลผล และในการทดสอบที่ 6 กับ 7 จะเป็นการทดสอบการคำนวณเพื่อให้ทราบถึงระยะเวลาที่ใช้ในการติดต่อสื่อสารระหว่างหน่วยประมวลผลกลางกับหน่วยประมวลผลกราฟิกส์ และระยะเวลาที่ใช้ในการคำนวณจริงบนหน่วยประมวลผลกราฟิกส์

#### 4.1.3 แนวคิดในการประเมินผล

โดยแนวคิดในการประเมินผลการทำงานของวิธีจัดการงานที่ได้คือ จะใช้โปรแกรมจำลองสัญญาณที่ได้ทำการปรับปรุงด้วยเทคนิคการทำงานแบบลูกผสมจากวิธีที่ได้ทดสอบบนระบบคลัสเตอร์ ตามข้อกำหนดในเรื่องของเวลาที่ใช้งานเปรียบเทียบกับโปรแกรมจำลองสัญญาณที่คำนวณแบบลำดับในด้านต่างๆ เช่น ประสิทธิภาพจากการแบ่งงาน ความคลาดเคลื่อนของผลลัพธ์ รวมทั้งเปรียบเทียบเวลาที่ใช้ในการคำนวณจริงกับเวลาที่คำนวณจากสูตรคำนวณเวลาที่ได้

### 4.2 การพัฒนาโปรแกรมคำนวณแบบขนานด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม

เพื่อที่โปรแกรมคำนวณแบบขนานจะสามารถใช้ประโยชน์จากหน่วยประมวลผลกราฟิกส์ที่มีอยู่โดยมอบหมายงานให้ไปคำนวณได้ตามแนวคิดในการทดสอบ จำเป็นที่จะต้องเขียนโค้ดเพิ่มในส่วนของการติดต่อกับหน่วยประมวลผลกราฟิกส์ ซึ่งในงานวิจัยนี้ระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้ในการทดลองใช้งานกราฟิกส์การ์ดที่ผลิตโดยบริษัท NVIDIA จึงเลือกใช้ CUDA เพื่อทำหน้าที่ช่วยติดต่อกับหน่วยประมวลผลกราฟิกส์ และใช้คอมไพเลอร์ nvcc ซึ่งพัฒนาโดยบริษัทข้างต้นในการคอมไพล์ส่วนของโค้ดหลัก ซึ่งรายละเอียดของการพัฒนาจะมีดังนี้

#### 4.2.1 การพัฒนาส่วนโค้ดหลัก

จากโปรแกรมจำลองสัญญาณที่คำนวณแบบขนานเดิมซึ่งได้มอบหมายให้หน่วยประมวลผลกลางรับหน้าที่ในการคำนวณพื้นที่ในระดับความละเอียดต่างๆ นำมาสู่การพัฒนาเพื่อที่จะให้หน่วยประมวลผลกราฟิกส์ที่อยู่ในแต่ละเครื่องคำนวณสามารถแบ่งเบาภาระของงานได้ จึงต้องมีการแก้ไขปรับปรุงโค้ดส่วนที่เป็นการทำงานหลักให้สามารถติดต่อกับหน่วยประมวลผลกราฟิกส์ได้

เพื่อที่สามารถให้หน่วยประมวลผลกราฟิกส์รับส่งข้อมูลได้อย่างมีประสิทธิภาพ ในงานวิจัยนี้จะใช้เทคนิคที่มีชื่อว่า “Zero-Copy memory” ซึ่งจะสามารถลดเวลาในการสื่อสารได้ เนื่องจากเท



รตของหน่วยประมวลผลกราฟิกส์จะสามารถเข้าถึงหน่วยความจำหลักได้โดยตรง โดยเรียกใช้งานคำสั่ง `cudaHostGetDevicePointer` เพื่อให้ตัวแปรของหน่วยประมวลผลกราฟิกส์สามารถทราบถึงตำแหน่งและเข้าถึงหน่วยความจำหลักได้โดยตรง และการเรียกใช้งานส่วนการทำงานบนหน่วยประมวลผลกราฟิกส์ซึ่งจะตามด้วยสัญลักษณ์ “`<<< , >>>`” เพื่อกำหนดจำนวนของเทรตที่จะใช้งาน ดังรูปที่ 4-1 และยังสามารถเรียกชุดคำสั่งแบบปกติหลังได้หลังจากที่มีการเรียกใช้เคอเนล เนื่องจากภายหลังการเรียกใช้คำสั่งเคอเนลแล้วการควบคุมทั้งหมดจะเป็นหน้าที่ของหน่วยประมวลผลกลาง ซึ่งถ้ากำหนดปริมาณงานที่เหมาะสมก็จะสามารถลดเวลาในการทำงานได้ โดยส่วนของการทำงานที่จำถูกปรับปรุงจะได้แก่ INTERQT, NLMAS, NLMMT1, NLMMT2, ETAMAP เนื่องจากเป็นส่วนการทำงานที่เกี่ยวกับการอนุรักษ์มวลและโมเมนตัมซึ่งใช้เวลาค่อนข้างมากดังตารางที่ 4-1

ตารางที่ 4-1 เวลาที่ใช้ของแต่ละส่วนการทำงานและเวลาที่ใช้ในการทำงานทั้งหมดของโปรแกรม

TUNAMI		
36000 steps	3-node Cluster (sec.)	8-node Cluster (sec.)
interqt(Region1)	113.426	190.091
nlmass(Region2)	193.679	239.308
nlmmt1(Region2)	248.184	435.339
nlmmt2(Region2)	2381.398	3264.331
etamap(Region2)	42.045	53.276
Execution time	3700.015	4870.426

```

float *d_ma, *d_mb;

float *h_ma, *h_mb;

//----- Send Host Pointer -----//

cudaHostGetDevicePointer((void **)&d_ma, (void *) d_ma, 0);

cudaHostGetDevicePointer((void **)&d_mb, (void *) d_mb, 0);

//----- Call Kernel -----//

```

รูปที่ 4-1 ตัวอย่างการเรียกใช้คำสั่งเพื่อคำนวณงานบนหน่วยประมวลผลกราฟิกส์

#### 4.2.2 การคอมไพล์โปรแกรมแบบขนานลูกผสม

เนื่องจากโปรแกรมจำลองสึนามิที่เป็นพื้นฐานในการพัฒนาของโปรแกรมแบบขนานลูกผสมนี้ ได้ใช้งานไลบรารีของ MPICH เพื่อสามารถควบคุมการใช้งานเครื่องสำหรับประมวลผลต่างๆที่อยู่ในระบบ แต่การพัฒนาโปรแกรมที่เพิ่มเติมเข้ามาจะเป็นส่วนของการเรียกใช้งานหน่วยประมวลผลกราฟิกส์เพื่อประมวลผล ซึ่งจะต้องใช้งานไลบรารีของ CUDA จะพบว่าทั้งสองกรณีนี้ความจริงแล้วจะต้องคอมไพล์ด้วยคอมไพเลอร์ที่แตกต่างกันเพื่อใช้งานโปรแกรมในแต่ละส่วน แต่ด้วยความสามารถของคอมไพเลอร์ซึ่งสามารถรวบรวมไลบรารีได้หลากหลายจึงทำให้ในงานวิจัยนี้คอมไพล์โปรแกรมที่คำนวณแบบขนานลูกผสมได้ง่ายยิ่งขึ้นดังรูปที่ 4-2

จากรูปที่ 4-2 จะแสดงให้เห็นถึงการคอมไพล์ของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสมทั้งหมด โดยจะคอมไพล์ด้วยคอมไพเลอร์ nvcc และนำเข้าไลบรารีของ MPICH เพื่อใช้งานในส่วนที่เป็น Message Passing

```
[champpi@frontend-0-0 ptunami]$ cat compile.sh
#!/bin/bash
lib_MPICH="/share/apps/mpich/intel/lib"
inc_MPICH="/share/apps/mpich/intel/include"
inc_CUDA_COM="/opt/cuda/sdk/C/common/inc"
export MALLOC_CHECK_=0

if [ 'rcuda' = "$1" ];
then
/share/apps/mpich/intel/bin/mpif90 -c futils.f tunami.f
/opt/cuda/bin/nvcc futils.o tunami.o xmlParser.cpp mpiutils.cpp main_cu.cu out.cpp parser.cu -o launch \
-I$inc_MPICH -L$lib_MPICH -lm -lmpich -lsvml -lifcore -lirc -limf -arch=sm_21 -DPROFILE $BIN -I$inc_CUDA_COM
fi
[champpi@frontend-0-0 ptunami]$
```

#### รูปที่ 4-2 ตัวอย่างการคอมไพล์โปรแกรมแบบขนานลูกผสม

#### 4.2.3 การเรียกใช้งานโปรแกรมแบบขนานลูกผสม

ในการเรียกใช้งานโปรแกรมแบบขนานลูกผสมที่ได้ทำการพัฒนาขึ้น เนื่องจากเป้าหมายคือ การเรียกใช้งานในรูปแบบ Message Passing ดังนั้นจึงสามารถเรียกด้วยคำสั่ง mpirun ได้เหมือนกับการคอมไพล์ด้วย MPICH ทั่วไป

```
$MPICH_HOME/bin/mpirun -number_of_process $NSLOTS -machinefile $TMPDIR/machines $MPICH_APP
```

#### รูปที่ 4-3 ตัวอย่างการเรียกใช้งานโปรแกรมแบบขนานลูกผสม

#### 4.3 การวิเคราะห์ประสิทธิภาพของโปรแกรมที่คำนวณแบบขนาน

ในการจะหาความเหมาะสมของรูปแบบของข้อมูลจำเป็นต้องอาศัยการศึกษาปัจจัยหลักที่ส่งผลกระทบต่อประสิทธิภาพของโปรแกรมคำนวณแบบขนาน ตามแนวคิดของงานวิจัยซึ่งจะประกอบด้วย

##### 4.3.1 การวิเคราะห์ประสิทธิภาพเชิงเวลาของแต่ละส่วนการทำงานของโปรแกรม

การศึกษาลักษณะการทำงานของโปรแกรม จำเป็นต้องเก็บข้อมูลการทำงานในแต่ละส่วนของโปรแกรม เพื่อให้ทราบถึงเวลาที่ใช้และประสิทธิภาพของเครื่องที่ใช้งาน อีกทั้งเพื่อให้ได้ความสัมพันธ์ของเวลาที่ใช้กับลักษณะการทำงานต่างๆของโปรแกรม โดยจะทำการวิเคราะห์จากโปรแกรมจำลองสี่มิติทั้งแบบการคำนวณแบบขนานและแบบขนานลูกผสม บนระบบเพกาซัสคลัสเตอร์ (Pegasus Cluster)

#### 4.3.1.1 การวิเคราะห์โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานและขนาน

##### ลูกผสม

สำหรับโปรแกรมจำลองสึนามิที่ใช้ในการทดลองในงานวิจัยนี้จะเก็บข้อมูลแต่ละส่วนการทำงานของโปรแกรมด้วยการแทรกคำสั่ง MPI\_Wtime ลงไประหว่างแต่ละส่วนการทำงานของโปรแกรมที่สนใจ

ตั้ง

รูปที่ 4-4

```
double stime, etime, ftime;
#ifdef __T_TIMING__
    stime = MPI_Wtime();
#endif
// CALL SUBROUTINE
#ifdef __T_TIMING__
    etime = MPI_Wtime();
    ftime = ftime + etime - stime;
#endif
```

รูปที่ 4-4 การเก็บข้อมูลการทำงานของโปรแกรมด้วยการแทรกคำสั่ง MPI\_Wtime

ซึ่งส่วนการทำงานของโปรแกรมที่จะทำการเก็บข้อมูลในแต่ละระดับความละเอียดจะเป็นดังนี้

1. ระดับความละเอียด R1
  - INITIAL, S\_BARRIER
  - send1, send2, send3
  - wait1, wait2, wait3
  - recv1, recv2
  - preMASS, postMASS, preMOMENT, postMOMENT
  - AFDEFORM, OPENBOUND, INTERQT
  - ETAMAP, CHANGE, MAX, OUTB
  - OUTMAX, E\_BARRIER, W\_BARRIER
2. ระดับความละเอียด R2
  - INITIAL, S\_BARRIER

- send1, send2, send3, send4, send5
  - wait1, wait2, wait3, wait4, wait5
  - recv1, recv2, recv3, recv4, recv5
  - JNQ\_S2C
  - preNLMASS, postNLMASS
  - preNLMMT1, postNLMMT1, preNLMMT2, postNLMMT2
  - ETAMAP, CHANGE, MAX, OUTB
  - OUTMAX, E\_BARRIER, W\_BARRIER
3. ระดับความละเอียด R3
- INITIAL, S\_BARRIER
  - send1, send2, send3, send4, send5, send6
  - wait1, wait2, wait3, wait4, wait5, wait6
  - recv1, recv2, recv3, recv4, recv5, recv6
  - JNZ, JNQ
  - preNLMASS, postNLMASS
  - preNLMMT1, postNLMMT1, preNLMMT2, postNLMMT2
  - ETAMAP, CHANGE, MAX, OUTB
  - OUTMAX, E\_BARRIER, W\_BARRIER
4. ระดับความละเอียด R4
- INITIAL, S\_BARRIER
  - send1, send3, send4, send5
  - wait1, wait3, wait4, wait5
  - recv1, recv2, recv3, recv4, recv5
  - JNZ, JNQ
  - preNLMASS, postNLMASS
  - preNLMMT1, postNLMMT1, preNLMMT2, postNLMMT2
  - ETAMAP, CHANGE, MAX, OUTB
  - OUTMAX, E\_BARRIER, W\_BARRIER

ซึ่งจากงานวิจัย “การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่มีหลายระดับความละเอียด : การณีศึกษาการจำลองสึนามิ” ได้ทำการจัดกลุ่มของส่วนการทำงานโปรแกรม TUNAMI ซึ่งในงานวิจัยนี้จะนำผลของการจัดกลุ่มดังกล่าวมาใช้ด้วยดังนี้

1. ส่วนที่ไม่สามารถลดเวลาการทำงานได้จากการเพิ่มหน่วยประมวลผล
  - INITIAL, AFDEFORM, OPENBOUNDARY
  - JNQ\_S2C, JNZ, JNQ
  - CHANGE
2. ส่วนที่สามารถลดเวลาการทำงานได้จากการเพิ่มหน่วยประมวลผล
  - preMASS, postMASS
  - preMOMENT, postMOMENT
  - INTERQT
  - preNLMASS, postNLMASS
  - preNLMMT1, postNLMMT1
  - preNLMMT2, postNLMMT2
  - ETAMAP, MAX
  - OUTB, OUTMAX
3. ส่วนการรับส่งข้อมูลกันระหว่างหน่วยประมวลผลที่ใช้เวลาเพิ่มขึ้น เมื่อใช้จำนวนหน่วยประมวลผลมากขึ้น
  - send, wait, recv
4. ส่วนที่ต้องรอการทำงานกันระหว่างหน่วยประมวลผลเพื่อเริ่มการทำงานขั้นต่อไปพร้อม ๆ กัน
  - S\_BARRIER, W\_BARRIER, E\_BARRIER

ซึ่งในส่วนที่สามารถลดเวลาการทำงานได้จากการเพิ่มหน่วยประมวลผล และส่วนการรับส่งข้อมูลกันระหว่างหน่วยประมวลผลที่ใช้เวลาเพิ่มขึ้น เมื่อใช้จำนวนหน่วยประมวลผลมากขึ้น จำเป็นต้องนำมาวิเคราะห์หาความสัมพันธ์ของเวลาตามลักษณะของส่วนการทำงานนั้นๆ

ในเรื่องการเก็บข้อมูลส่วนของการทำงานจะใช้ชุดคำสั่งจากโปรแกรม IPM โดยเพิ่มเข้าไปในขั้นตอนการคอมไพล์โปรแกรม และหลังจากการเก็บข้อมูลการทำงานเสร็จเรียบร้อยแล้ว ก็จะนำมาคิดเป็นค่าถ่วงน้ำหนักสำหรับการทำงานของข้อมูลในระดับความละเอียดต่างๆ ที่จะส่งผลให้การจะได้รับหน่วยประมวลผลนั้นจะแตกต่างกันขึ้นกับระดับความละเอียดของปัญหา

#### 4.3.2 การวิเคราะห์ประสิทธิภาพเชิงเวลาที่ขึ้นกับขนาดของปัญหา

ตารางที่ 4-2 ขนาดของปัญหาในแต่ละระดับความละเอียดทั้งหมด

Region levels	Region ID	No. of Grid		Grid size (Units)	Sea area (units)	Region levels	Region ID	No. of Grid		Grid size (Units)	Sea area (units)
		X	Y					X	Y		
R1	1	690	840	579,600	449,678	R4	2122	679	745	505,855	440,115
R2	21	721	841	606,361	547,045		2131	595	601	357,595	246,310
	22	721	841	606,361	496,667		2141	379	637	241,423	154,148
	23	721	841	606,361	299,396		2151	271	817	221,407	125,652
	211	241	325	78,325	67,620		2152	739	493	364,327	152,237
R3	212	637	469	298,753	275,475		2153	523	529	276,667	131,121
	213	397	211	83,767	60,977		2211	703	457	321,271	182,585
	214	241	229	55,189	41,759		2212	703	493	346,579	190,094
	215	469	697	326,893	187,636		2213	703	421	295,963	146,077
	221	639	487	311,193	234,716		2311	667	457	304,819	107,779
	231	655	466	305,230	90,912		2312	631	457	288,367	140,935
	232	637	466	296,842	210,574		2313	703	421	295,963	103,332
	2111	487	325	158,275	131,692		2321	703	457	321,271	249,792
R4	2112	487	421	205,027	179,132		2322	631	349	220,219	107,433
	2121	643	469	301,567	248,562		2323	631	493	311,083	250,013

เมื่อปัญหาในการทำงานที่แตกต่างกันดังตารางที่ 4-2 ขนาดของปัญหาส่งผลต่อเวลาในการทำงานของแต่ละระดับความละเอียด ซึ่งจากงานวิจัย “การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่มีหลายระดับความละเอียด : การศึกษาการจำลองสึนามิ” ได้ทำการจัดกลุ่มความสัมพันธ์กับขนาดของปัญหาออกเป็นดังนี้

1. ส่วนการทำงานที่มีประสิทธิภาพเชิงเวลาขึ้นกับขนาดของปัญหาทั้งหมด (Grid size) ของแต่ละระดับความละเอียด คือ คิดทั้งส่วนของพื้นน้ำและพื้นดิน ซึ่งประกอบด้วยส่วนการทำงาน INITIAL, MAX, ETAMAP, CHANGE, OUTB, OUTMAX
2. ส่วนการทำงานที่มีประสิทธิภาพเชิงเวลาขึ้นกับขนาดของปัญหาเฉพาะส่วนของพื้นน้ำ (Sea area) ของแต่ละระดับความละเอียด ซึ่งประกอบด้วยส่วนการทำงาน MASS, MOMENT, NLMASS, NLMMT
3. ส่วนการทำงานที่มีประสิทธิภาพเชิงเวลาขึ้นกับขนาดของปัญหาเฉพาะบริเวณขอบของปัญหาของแต่ละระดับความละเอียด ซึ่งประกอบด้วยส่วนการทำงาน OPEN BOUNDARY, INTERQT, JNQ\_S2C, JNZ, JNQ

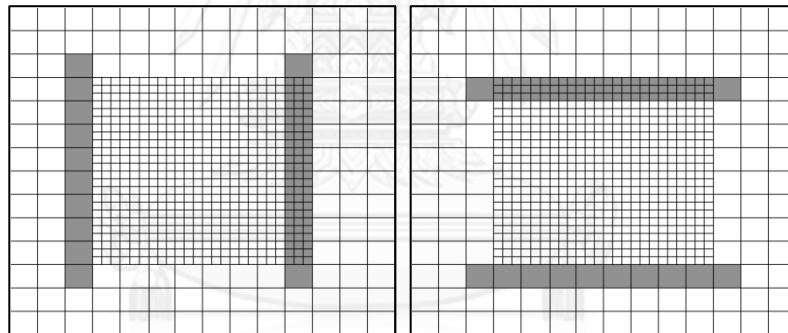
### 4.3.3 การวิเคราะห์การติดต่อสื่อสารระหว่างหน่วยประมวลผล

เนื่องจากลักษณะการทำงานของโปรแกรมที่การคำนวณมีความขึ้นต่อกันของข้อมูล (Data Dependency) จึงต้องมีส่วนที่รับส่งข้อมูลระหว่างหน่วยประมวลผล (Send และ Recv) โดยประสิทธิภาพเชิงเวลาจะขึ้นอยู่กับขนาดของปัญหาบริเวณขอบของปัญหาที่อยู่ติดกันในแต่ละระดับความละเอียดซึ่งสามารถจำแนกประเภทของการติดต่อสื่อสารได้เป็นดังนี้

#### 4.3.3.1 การแลกเปลี่ยนข้อมูลภายนอกของแต่ละระดับความละเอียด (Inter Communication)

การแลกเปลี่ยนข้อมูลภายนอกของแต่ละระดับความละเอียด เป็นการส่งข้อมูลบริเวณขอบที่อยู่ติดกัน ซึ่งเกิดจากความขึ้นต่อกันของข้อมูล ซึ่งลักษณะของการทำงานสามารถจำแนกได้ดังนี้

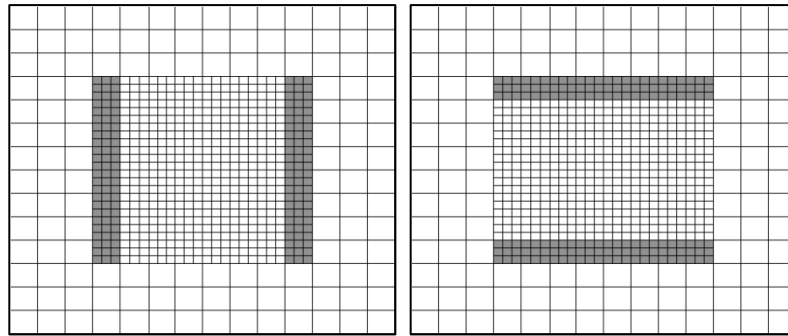
1. การส่งข้อมูลจากระดับ R1 ไประดับ R2
  - a.  $[m\_jmq, n\_jmq]$  – เป็นค่าตัวแปรที่ใช้สำหรับการทำงาน JNQ\_S2C



รูปที่ 4-5 บริเวณที่ระดับ R1 ต้องส่งค่าตัวแปร  $m\_jmq$  (ซ้าย) และค่าตัวแปร  $n\_jmq$  (ขวา) ให้ระดับ R2

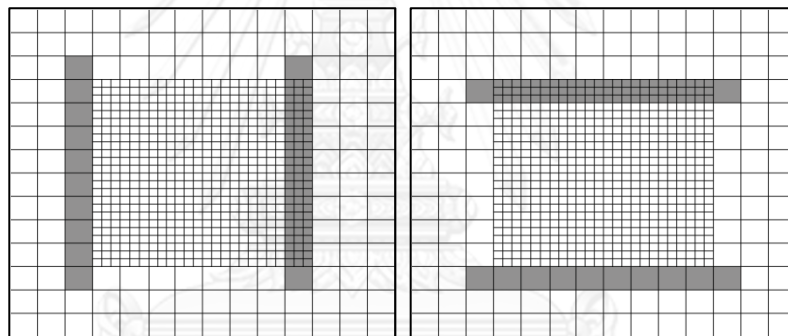
2. การส่งข้อมูลจากระดับ R2 ไประดับ R3 และการส่งข้อมูลจากระดับ R3 ไประดับ R4
  - a.  $[dzb]$  – เป็นค่าตัวแปรที่ใช้สำหรับการทำงาน JNZ





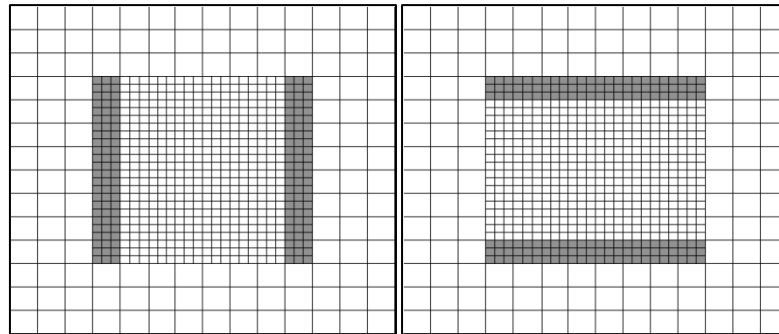
รูปที่ 4-6 บริเวณที่ระดับ R2 และ R3 ต้องส่งค่าตัวแปร dzb  
ให้ระดับ R3 และ R4

- b. [mb, nb] – เป็นค่าตัวแปรที่ใช้สำหรับการทำงาน JNQ



รูปที่ 4-7 บริเวณที่ระดับ R2 และ R3 ต้องส่งค่าตัวแปร mb (ซ้าย) และ  
nb (ขวา) ให้ระดับ R3 และ R4

3. การส่งข้อมูลจากระดับ R4 ไประดับ R3 และการส่งข้อมูลจากระดับ R3 ไประดับ R2
- a. [dzb, zb] – เป็นค่าตัวแปรที่ได้จากการทำงาน JNZ

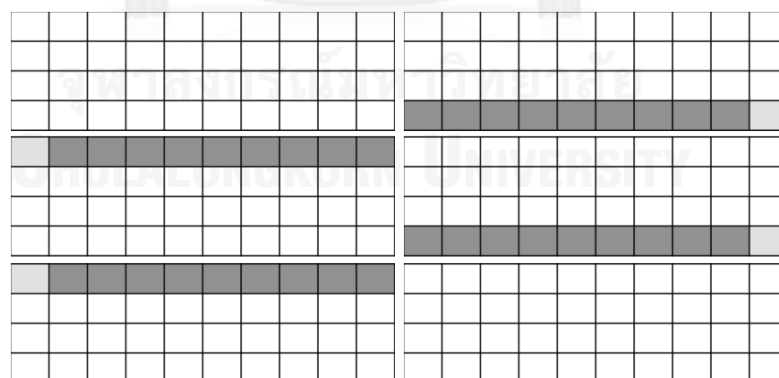


รูปที่ 4-8 บริเวณที่ระดับ R3 และ R4 ต้องส่งค่าตัวแปร dzb และ  
ค่าตัวแปร zb ให้ระดับ R2 และ R3

#### 4.3.3.2 การแลกเปลี่ยนข้อมูลภายในของแต่ละระดับความละเอียด (Intra Communication)

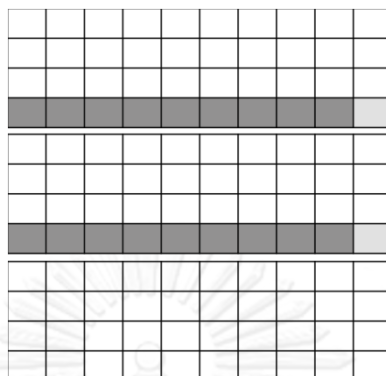
การแลกเปลี่ยนข้อมูลภายในของแต่ละระดับความละเอียด เป็นการส่งข้อมูลบริเวณขอบที่อยู่ติดกันซึ่งเกิดจากการแบ่งพื้นที่ในระดับความละเอียดนั้นๆ ออกเป็นหลายส่วนเพื่อลดขนาดของปัญหาในการทำงาน ซึ่งสามารถแบ่งออกได้เป็น 3 ส่วน ได้แก่ ส่วนล่าง (ส่วนที่ 0) ส่วนกลาง (ส่วนที่ 1 ถึง n-1) และส่วนบน (ส่วนที่ n-1) ซึ่งลักษณะของการทำงานสามารถจำแนกได้ดังนี้

1. การส่งข้อมูลระหว่างกันภายในระดับ R1
  - a. [ma, na] – เป็นตัวแปรที่ใช้สำหรับการทำงาน MASS



รูปที่ 4-9 บริเวณที่ส่วนย่อยในระดับ R1 ต้องส่งค่าตัวแปร ma (ซ้าย) และ  
ค่าตัวแปร na (ขวา) ให้ส่วนย่อยที่อยู่ติดกันในระดับ R1

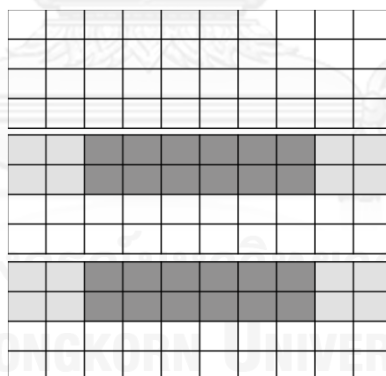
b. [zb] – เป็นตัวแปรที่ใช้สำหรับการทำงาน MOMENT



รูปที่ 4-10 บริเวณที่ส่วนย่อยในระดับ R1 ต้องส่งค่าตัวแปร zb ให้ส่วนย่อยที่อยู่ติดกันในระดับ R1

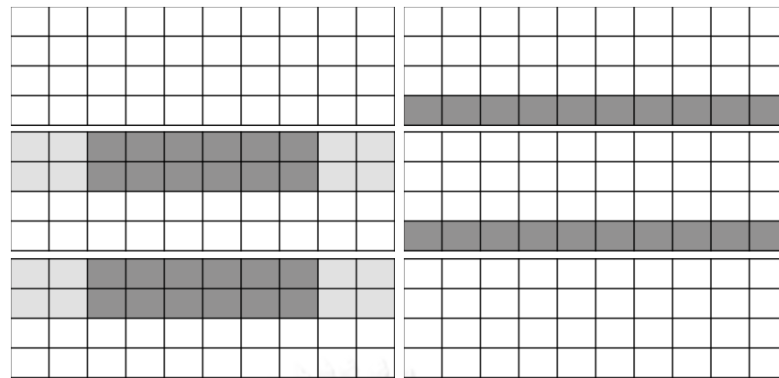
2. การส่งข้อมูลระหว่างกันภายในระดับ R2, R3 และ R4

a. [ma, na] – เป็นตัวแปรที่ใช้สำหรับการทำงาน NLMASS



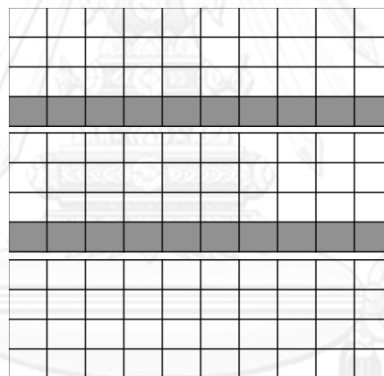
รูปที่ 4-11 บริเวณที่ส่วนย่อยในระดับ R2, R3, R4 ต้องส่งค่าตัวแปร ma และ na ให้ส่วนย่อยที่อยู่ติดกันในระดับ R2, R3, R4

b. [dzb, za, zb] – เป็นตัวแปรที่ใช้สำหรับการทำงาน NLMMT1



รูปที่ 4-12 บริเวณที่ส่วนย่อยในระดับ R2, R3, R4 ต้องส่งค่าตัวแปร dzb (ซ้าย) และค่าตัวแปร za, zb (ขวา) ให้ส่วนย่อยที่อยู่ติดกันในระดับ R2, R3, R4

c. [dma, dna] – เป็นตัวแปรที่ใช้สำหรับการทำงาน NLMMT2



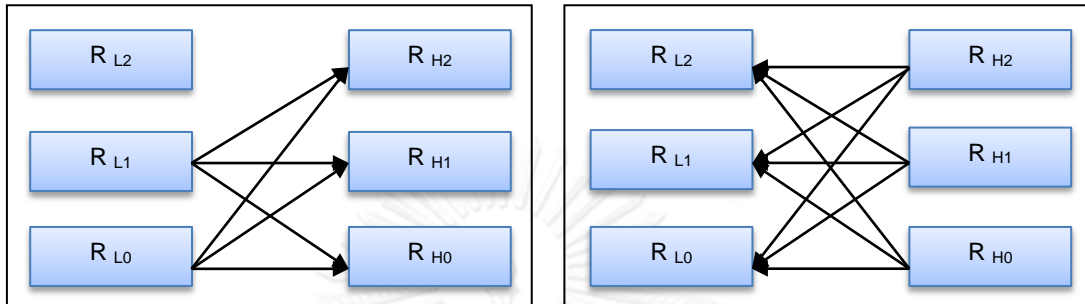
รูปที่ 4-13 บริเวณที่ส่วนย่อยในระดับ R2, R3, R4 ต้องส่งค่าตัวแปร dma และ dna ให้ส่วนย่อยที่อยู่ติดกันในระดับ R2, R3, R4



**4.3.3.3 การแลกเปลี่ยนข้อมูลระหว่างระดับความละเอียด เมื่อมีการแบ่งงานภายในระดับความละเอียด**

จากการที่แบ่งข้อมูลภายในระดับความละเอียด ทำให้ลักษณะของการแลกเปลี่ยนข้อมูลภายนอกความละเอียดนั้นแตกต่างกันตามขอบเขตที่อยู่ติดกันของแต่ละระดับความละเอียด ซึ่งสามารถแบ่งรูปแบบออกได้เป็น การส่งข้อมูลจากระดับที่มีความละเอียดต่ำกว่า ( $R_L$ ) ไปยังระดับที่มีความละเอียดสูงกว่า ( $R_H$ ) ซึ่งจะส่งข้อมูลจากระดับที่มีความละเอียดต่ำกว่าเฉพาะส่วนที่ติดกับระดับที่มีความละเอียดสูงกว่าเท่านั้น ไปยังทุกส่วนของระดับที่มีความละเอียดสูงกว่า ส่วนที่ไม่ติดกับระดับที่มีความละเอียดสูงกว่าจะไม่ส่งข้อมูล และการส่งข้อมูลจากระดับที่มีความละเอียดสูงกว่า ( $R_H$ ) ไปยัง

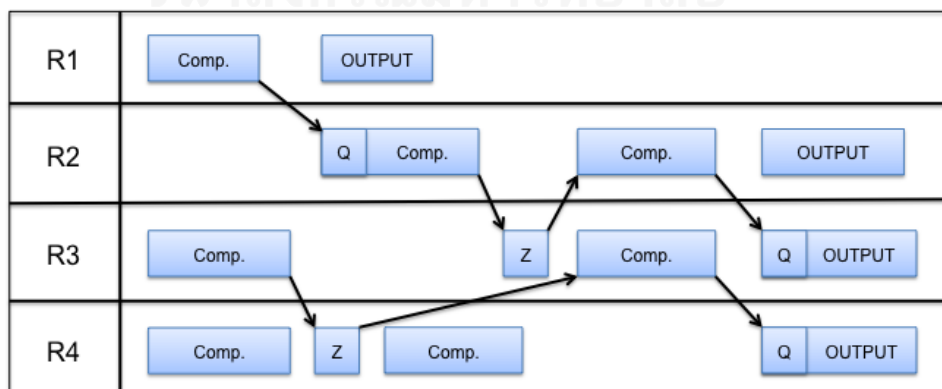
ระดับที่มีความละเอียดต่ำกว่า ( $R_L$ ) ซึ่งจะส่งข้อมูลจากระดับที่มีความละเอียดสูงกว่าทุกส่วนไปยังทุกส่วนของระดับที่มีความละเอียดต่ำกว่าดังรูปที่ 4-14



รูปที่ 4-14 การแลกเปลี่ยนข้อมูลภายนอกความละเอียด เมื่อมีการแบ่งงานภายในระดับความละเอียด

#### 4.3.3.4 การรองรับข้อมูลของการติดต่อสื่อสารระหว่างหน่วยประมวลผล

เนื่องจากลักษณะการทำงานของโปรแกรมที่มีความขึ้นต่อกันของข้อมูล ส่งผลให้เมื่อการทำงานใดๆต้องการเริ่มต้นทำงานแต่ข้อมูลยังไม่เพียงพอ จึงจำเป็นที่จะต้องมีการรองรับข้อมูลจากหน่วยประมวลผลอื่น เพื่อลดระยะเวลาในการรอข้อมูลดังกล่าวจึงต้องทำการวิเคราะห์ช่วงเวลาในการรองรับข้อมูลในแต่ละครั้งของการติดต่อสื่อสารดังรูปที่ 4-15 ซึ่ง Comp. แสดงถึงกระบวนการคำนวณตามหลักการอนุรักษ์มวลและโมเมนตัม และ OUTPUT แสดงถึงกระบวนการจัดการผลการคำนวณส่วน Q และ Z แสดงถึงส่วนการทำงานที่แลกเปลี่ยนข้อมูลระหว่างระดับความละเอียด

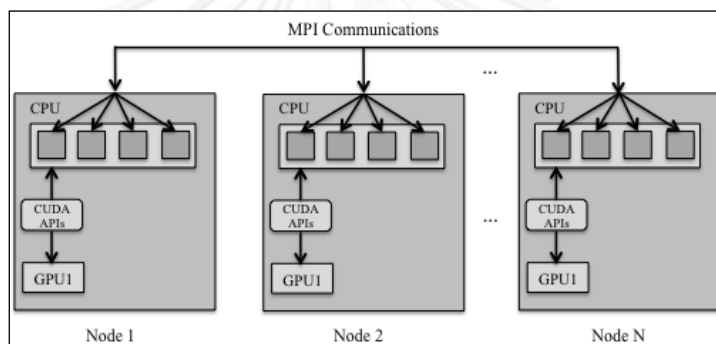


รูปที่ 4-15 การรองรับข้อมูลระหว่างหน่วยประมวลผล

#### 4.4 การแบ่งงานให้เหมาะสมบนระบบคลัสเตอร์คอมพิวเตอร์แบบไฮบริด

##### 4.4.1 การแบ่งงานให้เหมาะสมโดยปรับสัดส่วนของงานระหว่างหน่วยประมวลผล

ในการพัฒนาโปรแกรมเพื่อที่จะสามารถคำนวณงานบนหน่วยประมวลผลกราฟิกส์ได้แล้วนั้น ยังจำเป็นต้องหาสัดส่วนที่เหมาะสมสำหรับงานที่จะถูกกำหนดให้กับทั้งหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์อีกเช่นเดียวกัน เพื่อที่จะสามารถทำการคำนวณแบบขนานได้อย่างเหมาะสม โดยแผนภาพสำหรับโปรแกรมแบบขนานลูกผสมที่พัฒนาจะปรากฏดังรูปที่ 4-16 ซึ่งในแต่ละโหนดของระบบคลัสเตอร์หน่วยประมวลผลกราฟิกส์จะสามารถติดต่อกับหน่วยประมวลผลกลางได้เพียงหนึ่งตัวเท่านั้น



รูปที่ 4-16 แผนภาพของโปรแกรมแบบขนานลูกผสมสำหรับงานวิจัยนี้

โดยสมการที่จะแสดงต่อไปนี้ต้องการให้เวลาที่ใช้ในการทำงานของหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางนั้นเสร็จพร้อมกัน ซึ่งเวลาในการทำงานของหน่วยประมวลผลทั้งสองประเภทจะเกิดจากปริมาณที่ได้รับหารด้วยประสิทธิภาพในการทำงานของหน่วยประมวลผลนั้นๆ ( $T = W/P$ ) และเมื่อกำหนดให้เวลาของหน่วยประมวลผลทั้งสองประเภทนั้นเท่ากันจึงต้องมีสัดส่วนของงานที่หน่วยประมวลผลแต่ละประเภทนั้นจะได้รับไป ( $f$ )

แต่ทว่าการนำค่าประสิทธิภาพของหน่วยประมวลผลมาใช้ในการสมการเพื่อเป็นค่าสัดส่วนของการทำงานนั้นอาจจะไม่เหมาะสม เพราะฟังก์ชันที่ใช้ในการวัดผลของโปรแกรมทดสอบประสิทธิภาพนั้นแตกต่างจากฟังก์ชันที่ใช้ในการทำงานของโปรแกรมที่จะพัฒนา ในงานวิจัยนี้จึงเลือกใช้สัดส่วนของประสิทธิภาพ (Performance Ratio) ระหว่างโปรแกรมที่คำนวณแบบขนานกับโปรแกรมที่คำนวณแบบขนานลูกผสม ( $S = P_{GPU}/P_{CPU}$ )

เพื่อที่จะสามารถกำหนดปริมาณงานได้อย่างเหมาะสมตามแผนภาพด้านบนจะแสดงสมการดังต่อไปนี้ โดยกำหนดให้

- T แสดงถึงเวลาที่ใช้งานในการคำนวณของหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์
- P แสดงถึงประสิทธิภาพในการคำนวณของหน่วยประมวลผล
- W แสดงถึงปริมาณของงานที่ใช้ในการคำนวณทั้งหมด
- f แสดงถึงสัดส่วนของงานที่เหมาะสมเมื่อเวลาในการคำนวณของหน่วยประมวลผลกลางและหน่วยประมวลผลกราฟิกส์เท่ากัน
- n แสดงถึงจำนวนของหน่วยประมวลผลที่ไม่ได้ติดต่อกับหน่วยประมวลผลกราฟิกส์ (n = number of CPU – number of GPU)
- S แสดงถึงอัตราส่วนของประสิทธิภาพระหว่างหน่วยประมวลผลกราฟิกส์กับหน่วยประมวลผลกลาง (Performance Ratio) คูณด้วยจำนวนของหน่วยประมวลผลกราฟิกส์ (S = number of GPU \* Performance Ratio)

$$T_{GPU} = f \cdot \frac{W}{P_{GPU}}$$

$$T_{CPU} = (1 - f) \cdot \frac{W}{n \cdot P_{CPU}}$$

$$f \cdot \frac{W}{S} = (1 - f) \cdot \frac{W}{n}$$

สัดส่วนของปริมาณงานที่เหมาะสมจะได้แก่

$$f = \frac{S}{n + S}, 1 - f = \frac{n}{n + S}$$

ปริมาณงานที่เหมาะสมสำหรับหน่วยประมวลผลกลางที่ไม่ได้ติดต่อกับหน่วยประมวลผลกราฟิกส์ คือ

$$W_{CPU} = \frac{W}{n + S}$$

ปริมาณงานที่เหมาะสมสำหรับหน่วยประมวลผลกราฟิกส์ คือ

$$W_{GPU} = S \cdot W_{CPU}$$

#### 4.4.2 การจัดสรรงานให้เหมาะสมตามข้อกำหนดในเรื่องของเวลาและหน่วยประมวลผลที่ใช้งาน

ในการกำหนดจำนวนของหน่วยประมวลผลที่จะใช้คำนวณสำหรับแต่ละระดับความละเอียด จะได้มาจากการคำนวณ โดยต้องทราบค่าปริมาณงานเฉลี่ย (Average Job Size) ก่อนโดยคิดจากขนาดของปัญหาทั้งหมดหารด้วยจำนวนหน่วยประมวลผลทั้งหมดที่ใช้ในการทำงาน ซึ่งขนาดของปัญหาที่ระดับความละเอียด R1 ต้องคูณด้วยค่าสัดส่วนการทำงานของ R1 เทียบกับ R2 R3 R4 โดย

- ตัวแปร n คือ จำนวนของหน่วยประมวลผลกลางที่ใช้
- ตัวแปร k คือ ค่าสัดส่วนการทำงานของ R1 เมื่อเทียบกับ R2 R3 R4
- ตัวแปร  $S_1$  คือ สัดส่วนของประสิทธิภาพระหว่างหน่วยประมวลผลกราฟิกส์เทียบกับหน่วยประมวลผลกลาง ( $S_1 = P_{GPU}/P_{CPU}$ )
- ตัวแปร  $S_2$  คือ สัดส่วนของงานที่หน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์จะได้รับ

$$\text{Average Job Size} = \frac{\text{Total Job Size}}{\text{Total Equivalent CPU cores}}$$

$$= \frac{AS_{R1}(k) + AS_{R2} + AS_{R3} + AS_{R4} + \dots + AS_{RN}}{\left[n - \left(\frac{n}{4}\right)\right] + \left[(S_1 + S_2) * \left(\frac{n}{4}\right)\right]}$$

จากนั้นจึงนำค่า Average Job Size มาหาจำนวนของหน่วยประมวลผลกลางที่ต้องใช้สำหรับทำงานในแต่ละระดับความละเอียด ด้วยการนำขนาดของปัญหาในแต่ละระดับความละเอียดหารด้วยค่า Average Job Size



$$p = \frac{AS_{RN}}{\text{Average Job Size}}$$

โดยค่า  $p$  คือ จำนวนของประมวลผลกลางที่ต้องใช้ในการคำนวณระดับความละเอียดนั้นๆ แต่แท้จริงแล้วจากรูปที่ 4-16 ทุกๆ 4 หน่วยประมวลผลกลางจะมี 1 หน่วยประมวลผลที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ จึงต้องทำการจัดเรียงขนาดของข้อมูลใหม่เพื่อให้ทราบถึงจำนวนของหน่วยประมวลผลที่ใช้งานจริง

โดยการแบ่งงานของโปรแกรมจำลองสีนามินี้ใช้วิธีการแบ่งงานตามแนวนอน จึงนำค่า  $p$  มารวกับจำนวนแถวของปัญหาแล้วจะได้ขนาดของปัญหาที่ถูกแบ่งงานออกมา (Equivalent Average Job Size)

$$\text{Equivalent Average Job Size} = A_p = \text{len}_x * \frac{\text{len}_y}{p}$$

เมื่อได้ค่า Equivalent Average Job Size ( $A_p$ ) ซึ่งเป็นขนาดของปัญหาที่ถูกแบ่งเรียบร้อยแล้ว ก็จะนำมาหาเวลาที่ใช้ของส่วนการทำงานต่างๆ ด้วยสมการเชิงเส้นของส่วนการทำงานนั้นๆ เพื่อใช้สำหรับการประเมินเวลาที่จะใช้ในการทำงานของโปรแกรมต่อไป

## บทที่ 5

### การทดลองและวิเคราะห์ผลการวิจัย

#### 5.1 การทดลอง

##### 5.1.1 รูปแบบการทดลอง

จากแนวคิดที่ได้กล่าวมาในบทที่ 3 ได้นำมาทดลองเพื่อเก็บข้อมูลการทำงานของโปรแกรม ซึ่งจะมีรายละเอียดดังต่อไปนี้

##### 5.1.1.1 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองสำหรับงานวิจัยนี้ ได้นำข้อมูลจริงสำหรับการจำลองสินามิบางส่วนมาเพื่อทดสอบ โดยจะนำมาเฉพาะค่าที่จำเป็นและมีรายละเอียดดังตารางที่ 5-1

ตารางที่ 5-1 พิกัดและขนาดของแต่ละพื้นที่ในแต่ละระดับความละเอียดที่ใช้ในการทดลอง  
ทั้งโปรแกรมที่คำนวณแบบขนานและแบบขนานลูกผสม

Region levels	Region ID	No. of Grid		Grid size (units)	Sea area (units)	The coordinate of grid on overlapped region			
						From		To	
		X	Y	X	Y	X	Y		
R1	1	690	840	579,600	449,678	-	-	-	-
R2	21	721	841	606,361	547,045	271	481	360	585
	22	721	841	606,361	496,667	271	587	360	691
R3	211	241	325	78,325	67,620	482	424	561	531
	212	637	469	298,753	275,475	350	612	561	767
	221	639	487	311,193	234,716	430	6	642	167
R4	2111	487	325	158,275	131,692	68	68	229	175
	2112	487	421	205,027	179,132	68	180	229	319

	2121	643	469	301,567	248,562	400	56	613	211
	2211	703	457	321,271	182,585	403	330	636	481

สำหรับชุดข้อมูลที่จะใช้ในการทดลองจะปรากฏดังตารางที่ 5-2 ซึ่งจะแสดงชื่องานและจำนวนของหน่วยประมวลผลที่ใช้งาน โดยอักษร z ระบุถึงจำนวนโหนดที่คำนวณ อักษร x ระบุถึงจำนวนหน่วยประมวลผลที่ใช้ในการคำนวณต่อหนึ่งระดับความละเอียด และอักษร p ระบุถึงลำดับของชุดข้อมูลที่ใช้ในการทดสอบซึ่งมีโหนดที่แตกต่างกัน

ตารางที่ 5-2 ชื่องานและจำนวนของหน่วยประมวลผลที่ใช้งานของแต่ละชุดข้อมูลที่ใช้ในการทดลอง โปรแกรมจำลองสึนามิทั้งที่คำนวณแบบขนานและแบบขนานลูกผสม

Region ID (Number of region)	Number of processor per region			
	1	2	3	4
2111 (4)	z1-1x-p1 (4)	z1-2x-p1 (8)	z1-3x-p1 (12)	z1-4x-p1 (16)
2112 (4)	z1-1x-p2 (4)	z1-2x-p2 (8)	z1-3x-p2 (12)	z1-4x-p2 (16)
2121 (4)	z1-1x-p3 (4)	z1-2x-p3 (8)	z1-3x-p3 (12)	z1-4x-p3 (16)
2211 (4)	z1-1x-p4 (4)	z1-2x-p4 (8)	z1-3x-p4 (12)	z1-4x-p4 (16)
2111/2112 (5)	z2-1x-p1 (5)	z2-2x-p1 (10)	z2-3x-p1 (15)	z2-4x-p1 (20)
2111/2121 (6)	z2-1x-p2 (6)	z2-2x-p2 (12)	z2-3x-p2 (18)	z2-4x-p2 (24)
2111/2211 (7)	z2-1x-p3 (7)	z2-2x-p3 (14)	z2-3x-p3 (21)	z2-4x-p3 (28)
2111/2112/2121 (7)	z3-1x-p1 (7)	z3-2x-p1 (14)	z3-3x-p1 (21)	z3-4x-p1 (28)
2111/2112/2211 (8)	z3-1x-p2 (8)	z3-2x-p2 (16)	z3-3x-p2 (24)	z3-4x-p2 (32)
2111/2121/2211 (9)	z3-1x-p3 (9)	z3-2x-p3 (18)	z3-3x-p3 (27)	z3-4x-p3 (36)
2111/2112/2121/2211 (10)	z4-1x (10)	z4-2x (20)	z4-3x (30)	z4-4x (40)

#### 5.1.1.2 เครื่องมือที่ใช้ในการทดลอง

เครื่องมือที่ใช้ในงานวิจัยนี้ ประกอบไปด้วย

1. โปรแกรมจำลองสึนามิที่คำนวณแบบลำดับ (TUNAMI Program)

2. โปรแกรมจำลองสึนามิที่คำนวณแบบขนาน (Parallel Tsunami Simulation Program)
3. MPICH เวอร์ชัน 1.2.7p1 และ CUDA เวอร์ชัน 4
4. โปรแกรมเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนาน IPM เวอร์ชัน 0.9.80 และ Ploticus เวอร์ชัน 2.4.1
5. ระบบเพกาซัสคลัสเตอร์ (Pegasus Cluster) – ระบบคลัสเตอร์ขนาด 8 โหนดๆละ 4 หน่วยประมวลผล รวมเป็น 32 หน่วยประมวลผล ทำงานด้วยระบบปฏิบัติการ Rocks Clusters 6.0.1 (Intel Xenon CPU L5520 2.27 GHz, NVIDIA Quadro FX 3800, Gigabit Network)
6. ระบบสเปซคลัสเตอร์ (Space Cluster) – ระบบคลัสเตอร์ขนาด 4 โหนดๆละ 4 หน่วยประมวลผล รวมเป็น 16 หน่วยประมวลผล ทำงานด้วยระบบปฏิบัติการ Rocks Cluster 5.4.2 (Intel Core i5 CPU 650 3.20 GHz, NVIDIA GeForce GT 430, Gigabit Network)

คุณสมบัติของหน่วยประมวลผลกราฟิกส์ที่ใช้ มีดังนี้

1. NVIDIA Quadro FX 3800
  - CUDA Capability : 1.3
  - Total amount of global memory : 1023 MBytes
  - 192 CUDA cores (24 Multiprocessors x 8 CUDA Cores/MP)
  - Support host page-locked memory mapping : Yes
2. NVIDIA Geforce GT 430
  - CUDA Capability : 2.1
  - Total amount of global memory : 1023 MBytes
  - 96 cores (2 Multiprocessors x 48 CUDA Cores/MP)
  - Support host page-locked memory mapping : Yes

## 5.1.2 ผลการทดลอง

### 5.1.2.1 การพัฒนาโปรแกรมคำนวณแบบขนานด้วยเทคนิคการเขียนโปรแกรมแบบ ลูกผสม

เนื่องด้วยโปรแกรมแบบขนานลูกผสมที่พัฒนาขึ้นจำเป็นต้องใช้ค่าสัดส่วนสำหรับการแบ่งภาระงานระหว่างหน่วยประมวลผลกลางกับหน่วยประมวลผลกราฟิกส์ จึงทำการทดสอบด้วยส่วนการทำงานแบบต่างๆบนระบบคลัสเตอร์คอมพิวเตอร์เพื่อให้ทราบถึงประสิทธิภาพที่แท้จริงดังตารางที่ 5-3 พบว่าเมื่อทำการรันโปรแกรมตัวอย่างของ NVIDIA เพื่อทดสอบประสิทธิภาพอันได้แก่ VectorAdd (โปรแกรมบวกชุดข้อมูล) และ FDTD3d (โปรแกรมคำนวณแบบ FDM) [6] ผลปรากฏว่าค่าความเร็วที่เพิ่มขึ้น (Speedup) ที่ได้รับจากการใช้หน่วยประมวลผลกราฟิกส์นั้นแตกต่างกันเนื่องจากรูปแบบของการคำนวณที่ต่างกัน ซึ่งเมื่อทำการทดสอบกับส่วนการทำงานของโปรแกรม TUNAMI ก็เช่นเดียวกัน พบว่าประสิทธิภาพที่ได้รับจากเปลี่ยนหน่วยประมวลผลนั้นก็ยังคงแตกต่างกัน งานวิจัยนี้จึงได้ทำการเลือกค่าสัดส่วนของประสิทธิภาพที่มากที่สุดของการทดสอบด้วยโปรแกรม TUNAMI สำหรับใช้ในการทดสอบการแบ่งงานของโปรแกรมแบบขนานลูกผสมในเบื้องต้น

ตารางที่ 5-3 ผลการทดสอบค่าสัดส่วนประสิทธิภาพระหว่างหน่วยประมวลผลกลางเทียบกับหน่วยประมวลผลกราฟิกส์เมื่อทดลองด้วยส่วนการทำงานแบบต่างๆบนคลัสเตอร์คอมพิวเตอร์

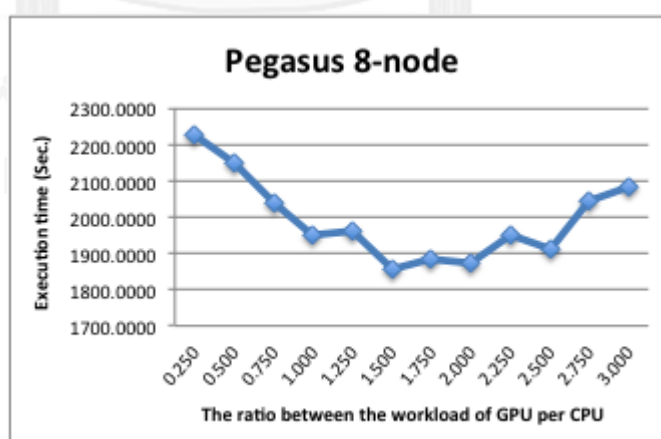
Benchmark	Pegasus (sec.)			Acer (sec.)		
	CPU	GPU	Speedup	CPU	GPU	Speedup
VectorAdd	0.6714	0.0348	<b>19.30</b>	1.0081	0.0867	<b>11.63</b>
FDTD3d	22.0000	0.1342	<b>163.88</b>	17.0000	0.3920	<b>43.37</b>
INTERQT	120.8549	100.0455	<b>1.21</b>	100.9050	52.4454	<b>1.92</b>
NLMAS	242.1596	282.2373	<b>0.86</b>	193.6794	92.4925	<b>2.09</b>
NLMMT1	412.8399	546.0846	<b>0.76</b>	248.1848	135.9917	<b>1.83</b>
NLMMT2	3411.9025	1937.4801	<b>1.76</b>	2381.3980	602.2757	<b>3.95</b>
ETAMAP	57.2211	44.3919	<b>1.29</b>	42.0454	38.6803	<b>1.09</b>

จากรูปที่ 5-1 และรูปที่ 5-2 จะแสดงถึงสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลาง ซึ่งตารางที่ 5-4 และตารางที่ 5-5 จะแสดงปริมาณของข้อมูลทำการทดลองทั้งระบบเพาซคลัสเตอร์และสเปซคลัสเตอร์ เมื่อใช้งานหน่วยประมวลผล 4 ตัวต่อ

หนึ่งพื้นที่ระดับความละเอียด และ ใช้งานหน่วยประมวลผล 2 ตัวต่อหนึ่งพื้นที่ระดับความละเอียด ตามลำดับ ซึ่งตารางจะแสดงปริมาณข้อมูลที่ถูกแบ่งออกเป็นสัดส่วนต่างๆจากปริมาณข้อมูลทั้งหมดที่ระดับความละเอียด R2 โดยรายละเอียดของตารางมีดังนี้

- SIZE คือปริมาณงานทั้งหมด
- GPU คือปริมาณงานที่หน่วยประมวลผลกราฟิกส์ได้รับไปคำนวณ
- CPU-C คือปริมาณงานที่หน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ได้รับไปคำนวณ
- CPU คือปริมาณงานที่หน่วยประมวลผลกลางตัวอื่นๆที่ไม่ได้ติดต่อกับหน่วยประมวลผลกราฟิกส์ได้รับไปคำนวณ
- G/C คือสัดส่วนของปริมาณงานหน่วยประมวลผลกราฟิกส์เทียบกับปริมาณงานหน่วยประมวลผลกลางปกติ

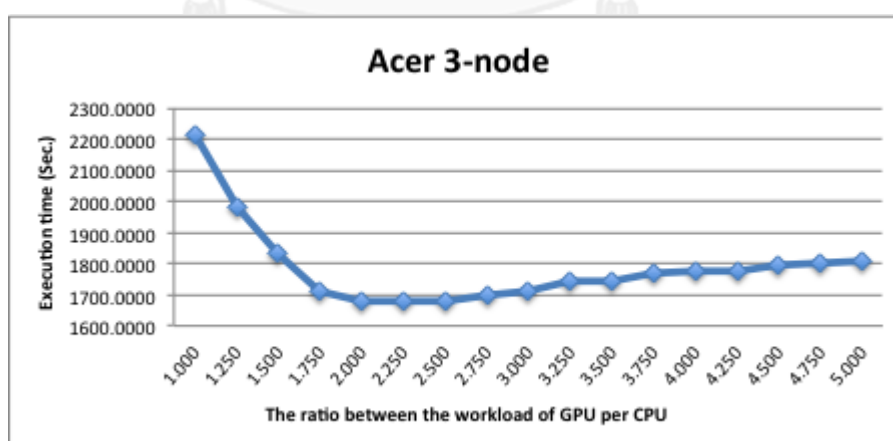
นอกจากนั้นในการเริ่มต้นกระบวนการหาสัดส่วนของงานที่เหมาะสมนั้นจะต้องทำการรันโปรแกรมในรูปแบบลำดับ และในรูปแบบขนานลูกผสมโดยใช้งานหนึ่งหน่วยประมวลผลต่อหนึ่งพื้นที่ระดับความละเอียด เพื่อเก็บข้อมูลเวลาที่ใช้ในการคำนวณเพื่อเป็นสัดส่วนเริ่มต้นในการทดสอบ โดยที่ระบบเพกาซัสสามารถวัดค่าได้ 1.750 และที่ระบบสเปซคลัสเตอร์สามารถวัดได้ 4.00 และสัดส่วนที่เหมาะสมที่ทดสอบได้จากระบบเพกาซัสคือ 1.500 และจากระบบเพกาซัสคือ 2.250



รูปที่ 5-1 ผลการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบเพกาซัสคลัสเตอร์

ตารางที่ 5-4 ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบเฟก้าซคลัสเตอร์

SIZE	GPU	CPU-C	CPU	G/C	#1	#2	#3
841	120	-	240	0.250	2226.2744	2298.6384	2226.2744
841	120	-	240	0.500	2148.4832	2232.0415	2148.4832
841	168	-	224	0.750	2037.3340	2037.3340	2090.4019
841	210	-	210	1.000	1952.0758	1952.0758	2027.4340
841	248	-	198	1.250	1959.7804	1959.7804	2065.7686
841	281	-	187	1.500	1854.8040	1854.8040	1856.3071
841	310	-	177	1.750	1886.2941	1935.8884	1886.2941
841	336	-	168	2.000	1870.2745	1935.4788	1870.2745
841	360	-	160	2.250	1951.2319	1951.2319	1956.9296
841	383	-	153	2.500	1913.5283	1960.2719	1913.5283
841	402	-	146	2.750	2046.2468	2087.4404	2046.2468
841	420	-	140	3.000	2082.7832	2082.7832	2100.1174



รูปที่ 5-2 ผลการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และหน่วยประมวลผลกลางที่ระบบสเปซคลัสเตอร์

ตารางที่ 5-5 ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์และ  
หน่วยประมวลผลกลางที่ระบบสเปคัลส์เตอร์

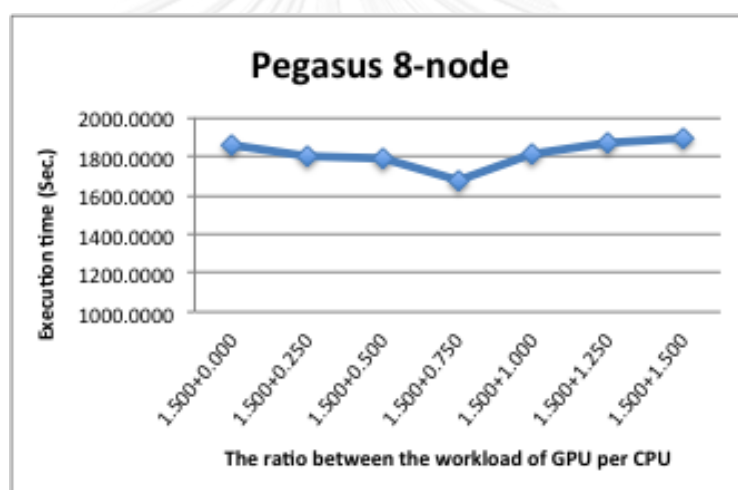
SIZE	GPU	CPU-C	CPU	G/C	#1	#2	#3
841	421	-	420	1.000	2214.1460	2215.8706	2214.1460
841	468	-	373	1.250	1981.0222	1989.8040	1981.0222
841	504	-	337	1.500	1836.9536	1836.9536	1861.3125
841	536	-	305	1.750	1708.4867	1721.2334	1708.4867
841	560	-	281	2.000	1681.0925	1681.0925	1684.5620
<b>841</b>	<b>583</b>	<b>-</b>	<b>258</b>	<b>2.250</b>	<b>1679.3641</b>	<b>1679.3641</b>	<b>1702.5433</b>
841	600	-	241	2.500	1681.3752	1681.3752	1699.0823
841	616	-	225	2.750	1695.9325	1695.9325	1699.8728
841	630	-	211	3.000	1711.2965	1711.2965	1737.3021
841	644	-	197	3.250	1743.4296	1745.7070	1743.4296
841	655	-	186	3.500	1744.8468	1758.8307	1744.8468
841	664	-	177	3.750	1771.4258	1771.4258	1779.5809
<b>841</b>	<b>672</b>	<b>-</b>	<b>169</b>	<b>4.000</b>	<b>1778.3480</b>	<b>1778.3480</b>	<b>1790.5541</b>
841	680	-	161	4.250	1772.8020	1793.2574	1772.8020
841	689	-	152	4.500	1793.0978	1793.0978	1798.4840
841	694	-	147	4.750	1804.5337	1806.4220	1804.5337
841	700	-	141	5.000	1811.0055	1811.0055	1811.0055

จากตารางที่ 5-4 และตารางที่ 5-5 จะพบว่าปริมาณงานที่คำนวณของหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ (CPU-C) นั้นยังไม่ได้ถูกกำหนด และเพื่อที่จะสามารถเพิ่มประสิทธิภาพให้กับโปรแกรมที่พัฒนาขึ้นได้จึงทำการทดลองปรับปรุงเพิ่มเติมในส่วนของ ประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ (CPU-C) ให้ได้รับงานในสัดส่วนที่แตกต่างกันเพื่อหาสัดส่วนของงานที่เหมาะสมที่จะกำหนดให้กับหน่วยประมวลผลข้างต้นได้



โดยขนาดของงานที่จะกำหนดเพิ่มเติมจำเป็นต้องไม่มากไปกว่าหน่วยประมวลผลกลางตัวอื่นๆ เพราะการทำงานของหน่วยประมวลผลกลางนี้จะอยู่ภายใต้กรอบเวลาของหน่วยประมวลผลกราฟิกส์ที่ทำงานอยู่ ก่อนที่หน่วยประมวลผลกราฟิกส์จะเรียกใช้งานคำสั่งเพื่อประสานการทำงานเทรตทั้งหมด

จากตารางที่ 5-6 และตารางที่ 5-7 G+C แสดงถึงปริมาณงานรวมที่หน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์จะได้รับ Optimal Ratio + CPU-C แสดงถึงสัดส่วนของงานที่เหมาะสมบวกกับสัดส่วนที่จะกำหนดให้กับหน่วยประมวลผลกลาง Sum Ratio แสดงถึงสัดส่วนรวมที่คอร์นั้นจะได้รับไป ซึ่งจากการทดลองพบว่าที่ระบบเพกาซัสคลัสเตอร์สามารถกำหนดงานเพิ่มได้เป็นค่าสัดส่วน 0.75 และที่ระบบสเปซคลัสเตอร์สามารถกำหนดงานเพิ่มได้เป็นค่าสัดส่วน 0.75

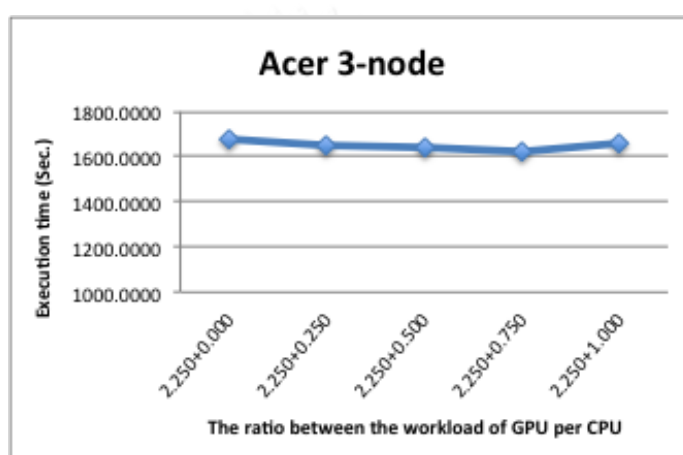


รูปที่ 5-3 ผลการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบเพกาซัสคลัสเตอร์

ตารางที่ 5-6 ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบเพกาซัสคลัสเตอร์

Size	G+C	G	CPU-C	CPU	Optimal Ratio + CPU-C	Sum Ratio	MIN.	#1	#2
841	281	281	0	187	1.500+0.000	1.500	1854.8040	1854.8040	1856.3071
841	310	266	44	177	1.500+0.250	1.750	1799.4875	1833.8120	1814.2986
841	336	252	84	168	1.500+0.500	2.000	1788.6414	1802.6495	1788.6414

841	360	240	120	160	1.500+0.750	2.250	1676.6893	1676.6893	1791.1071
841	383	230	153	153	1.500+1.000	2.500	1814.1676	1853.3457	1814.1676
841	402	219	182	146	1.500+1.250	2.750	1867.3541	1867.3541	1887.0763
841	420	210	210	140	1.500+1.500	3.000	1897.8365	1897.8365	2016.2109



รูปที่ 5-4 ผลการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบสเปซคลัสเตอร์

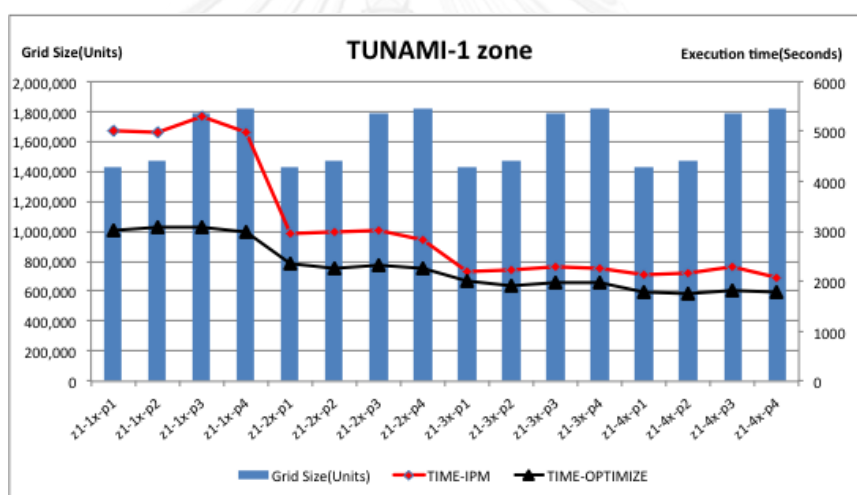
ตารางที่ 5-7 ตารางแสดงการหาสัดส่วนของงานที่เหมาะสมให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ที่ระบบสเปซคลัสเตอร์

Size	G+C	G	CPU-C	CPU	Optimal Ratio + CPU-C	Sum Ratio	MIN.	#1	#2
841	281	583	0	258	2.250+0.000	2.250	1679.3641	1679.3641	1702.5433
841	603	542.7	60	238	2.250+0.250	2.500	1649.3446	1649.3446	1649.4427
840	620	507.2727273	112	221	2.250+0.500	2.750	1636.6510	1636.6510	1639.4325
841	633	474.75	158	208	2.250+0.750	3.000	1619.4764	1619.4764	1628.9810
841	646	447.2307692	198	195	2.250+1.000	3.250	1657.2457	1657.2457	1670.4512

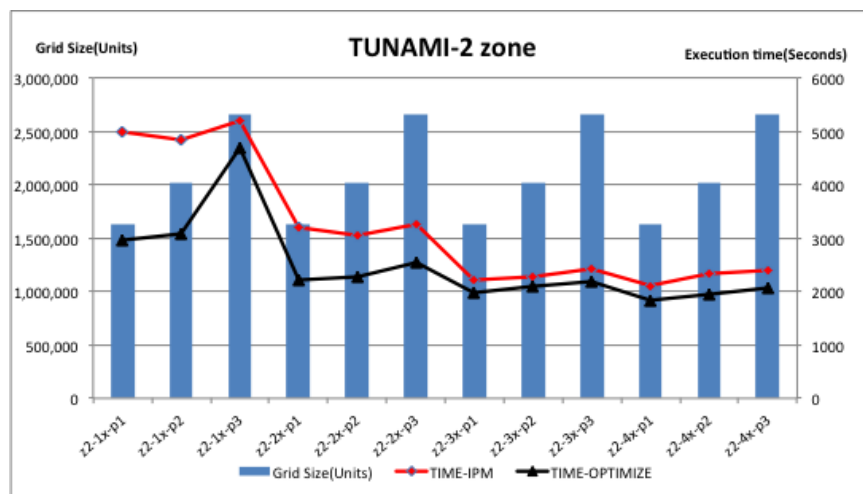
### 5.1.2.2 ผลการทำงานของโปรแกรมแบบขนานลูกผสม

ในการทดลองโปรแกรมจำลองสึนามิตามแนวคิดในการทดสอบข้อ 2-7 จะแสดงให้เห็นถึงความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดของปัญหาที่ถูกแบ่งออกตามจำนวนหน่วยประมวลผลที่ใช้คำนวณต่อหนึ่งพื้นที่ ซึ่งในการทดลองนี้จะเป็นผลจากระบบคลัสเตอร์เพกาซัสเท่านั้นเนื่องจากทรัพยากรของระบบสเปซคลัสเตอร์นั้นมีไม่เพียงพอต่อการทดลอง

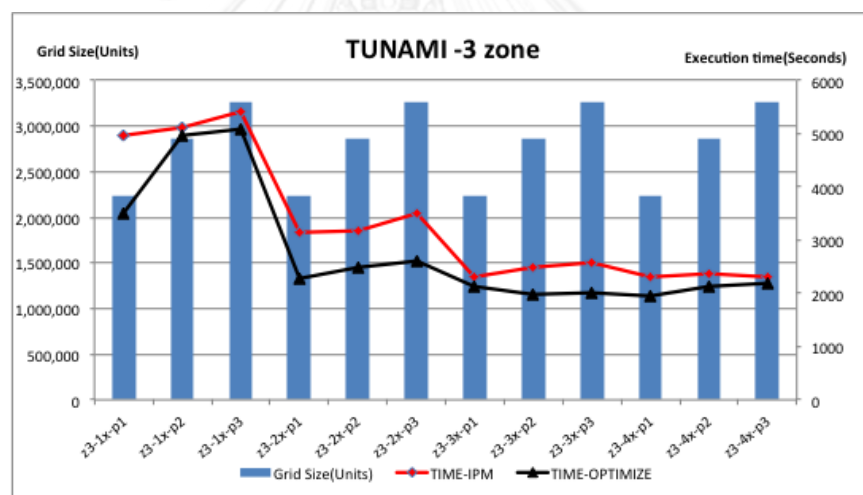
จากรูปที่ 5-5 ถึงรูปที่ 5-8 กราฟแท่งแสดงถึงขนาดของปัญหาทั้งหมดที่ใช้ในการคำนวณและกราฟเส้นสีแดงแสดงถึงเวลาที่ใช้ในการคำนวณของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานดั้งเดิม และกราฟเส้นสีดำแสดงถึงเวลาที่ใช้ในการคำนวณของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสมที่ได้พัฒนาขึ้นมาใหม่



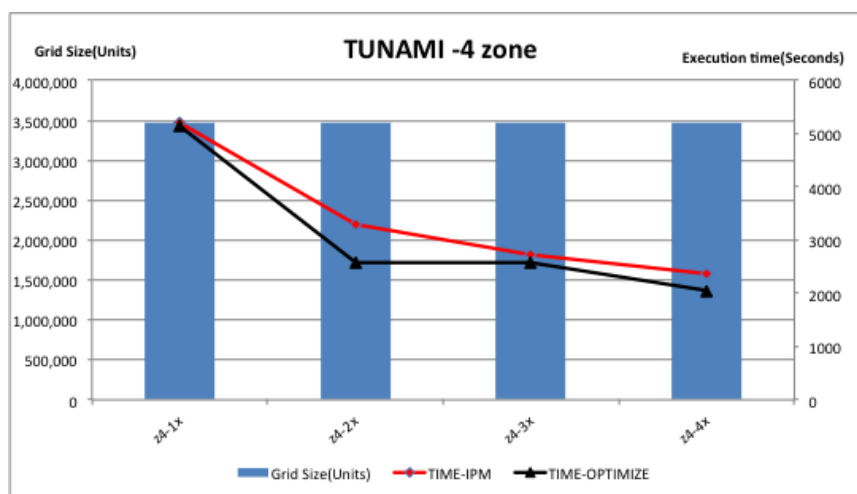
รูปที่ 5-5 ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบหนึ่งโชน



รูปที่ 5-6 ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบสองโชน



รูปที่ 5-7 ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบสามโชน



รูปที่ 5-8 ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดปัญหาของการคำนวณแบบสี่โซน

จากรูปที่ 5-5 ถึงรูปที่ 5-8 จะพบว่าการพัฒนาโปรแกรมด้วยเทคนิคแบบขนานลูกผสมสามารถลดเวลาในการคำนวณได้อย่างชัดเจน โดยความแตกต่างอย่างเห็นได้ชัดจะอยู่ที่การคำนวณแบบหนึ่งโซนเนื่องจากปริมาณหน่วยประมวลผลใช้งานเมื่อคิดเป็นสัดส่วนกับหน่วยประมวลผลกราฟิกส์นั้นยังค่อนข้างมาก แต่เมื่อเพิ่มจำนวนของหน่วยประมวลผลที่ใช้งานทั้งหมดโดยที่จำนวนของหน่วยประมวลผลกราฟิกส์ยังคงที่ ค่าประสิทธิภาพที่เพิ่มขึ้นของโปรแกรมจึงไม่สามารถแปรผันตามกับจำนวนของหน่วยประมวลผลที่เพิ่มขึ้นได้

## 5.2 การวิเคราะห์ผล

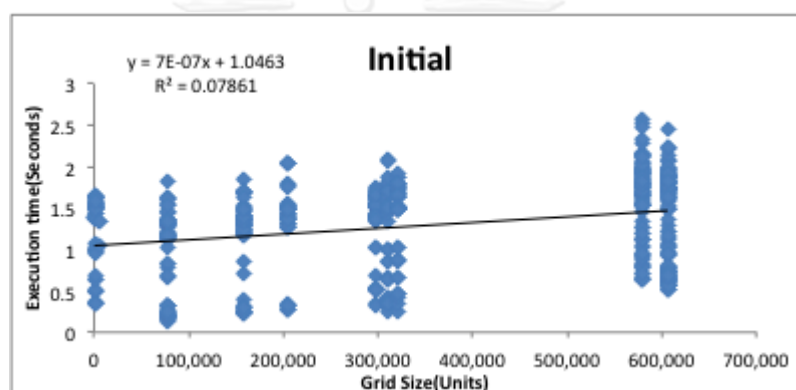
จากงานวิจัย “การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่มีหลายระดับความละเอียด : การณีศึกษาการจำลองสึนามิ” ได้จัดกลุ่มของส่วนการทำงานของโปรแกรมออกเป็นลักษณะต่างๆ ซึ่งงานวิจัยนี้จะวิเคราะห์ตามกลุ่มที่ได้ถูกสรุปมาเพื่อที่จะได้วิเคราะห์ไปในทิศทางเดียวกัน

### 5.2.1 การวิเคราะห์ผลโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม

ในการวิเคราะห์ประสิทธิภาพเชิงเวลากับขนาดของปัญหาตามส่วนการทำงานของโปรแกรมพบว่าสามารถจัดกลุ่มได้เป็นดังนี้

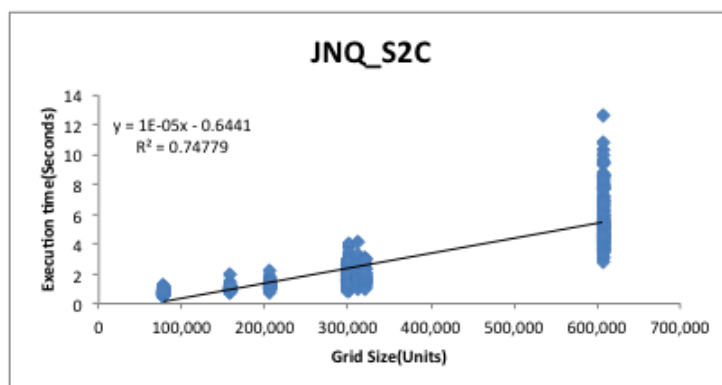
### 5.2.1.1 ส่วนการทำงานที่เวลาขึ้นกับขนาดปัญหาทั้งหมดในแต่ละระดับความละเอียด

ส่วนการทำงาน INITIAL สำหรับทุกระดับความละเอียดที่ใช้เวลาในการทำงานขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละระดับความละเอียด ตามรูปที่ 5-9 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของปัญหาทั้งหมดในแต่ละระดับความละเอียด ซึ่งจะกำหนดตัวแปรสำหรับใช้ในสมการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $A$  คือขนาดของปัญหาทั้งหมดในแต่ละระดับความละเอียด

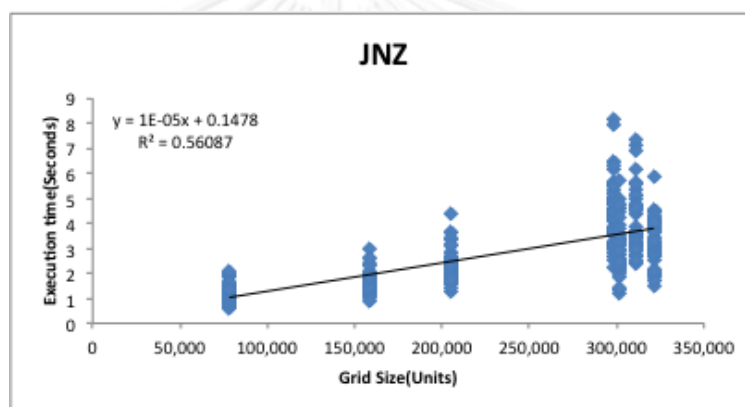


รูปที่ 5-9 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน INITIAL

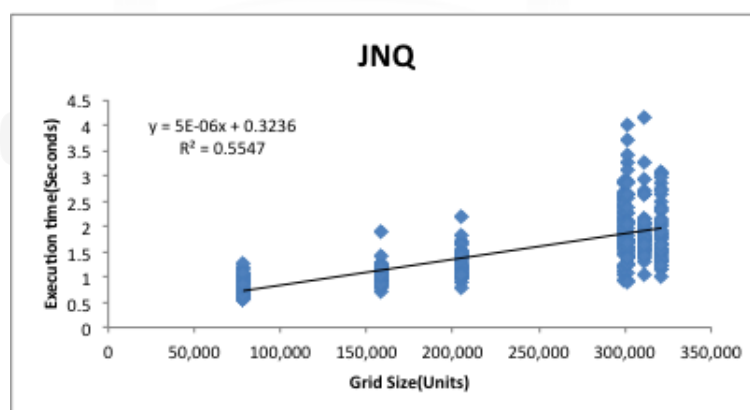
ส่วนการทำงาน JNQ\_S2C JNZ และ JNQ สำหรับระดับความละเอียด R2 R3 และ R4 ที่ใช้เวลาในการทำงานขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละระดับความละเอียด ตามรูปที่ 5-10 ถึงรูปที่ 5-12 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของปัญหาทั้งหมดในแต่ละระดับความละเอียด ซึ่งจะกำหนดตัวแปรสำหรับใช้ในสมการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $A$  คือขนาดของปัญหาทั้งหมดในแต่ละระดับความละเอียด



รูปที่ 5-10 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน JNQ\_S2C



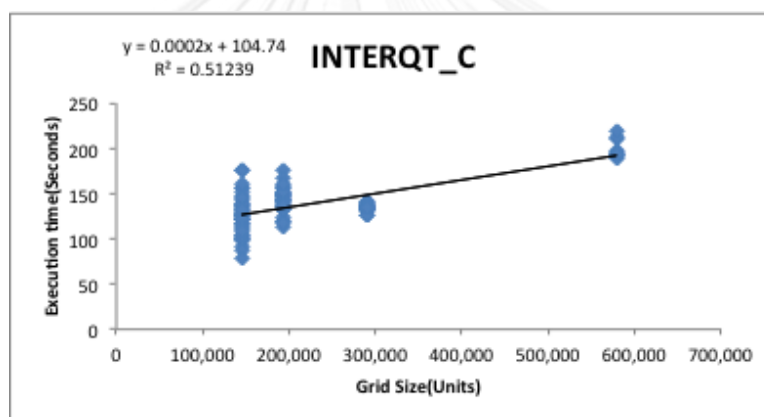
รูปที่ 5-11 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน JNZ



รูปที่ 5-12 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน JNQ

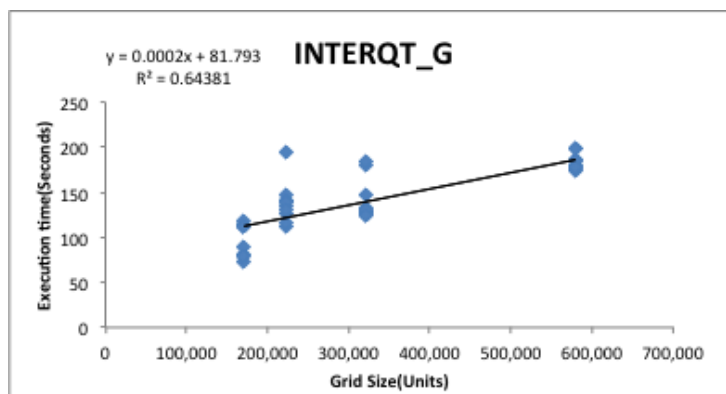
### 5.2.1.2 ส่วนการทำงานที่เวลาขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล

ส่วนการทำงาน INTERQT MAX และ ETACH (ETAMAP, CHANGE) สำหรับระดับความละเอียด R1 ที่ใช้เวลาในการทำงานขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล ซึ่งจะแสดงทั้งชื่อส่วนการทำงานที่ตามด้วย \_C คือใช้หน่วยประมวลผลกลางในการคำนวณ และ \_G คือใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ ตามรูปที่ 5-13 รูปที่ 5-10 ถึงรูปที่ 5-17 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล ซึ่งจะกำหนดตัวแปรสำหรับใช้ในสมการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $Ap$  คือขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล

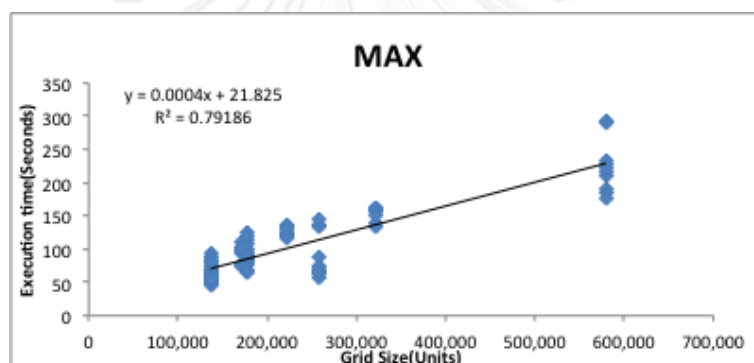


รูปที่ 5-13 ความสัมพันธ์ระหว่างเวลาที่ใช้กับขนาดของปัญหาของส่วนการทำงาน INTERQT ที่ใช้หน่วยประมวลผลกลางในการคำนวณ

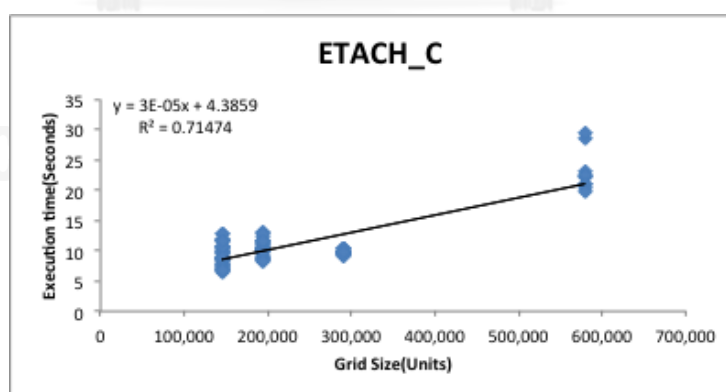




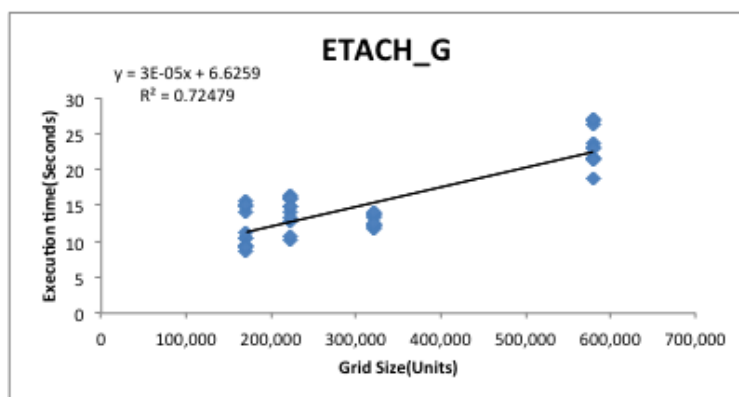
รูปที่ 5-14 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน INTERQT ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ



รูปที่ 5-15 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน MAX

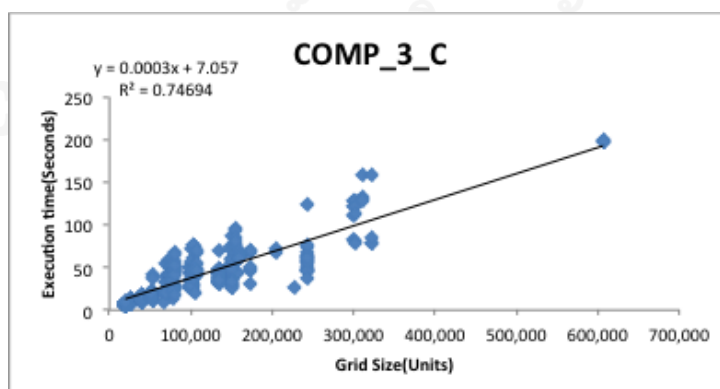


รูปที่ 5-16 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน ETACH ที่ใช้หน่วยประมวลผลกลางในการคำนวณ

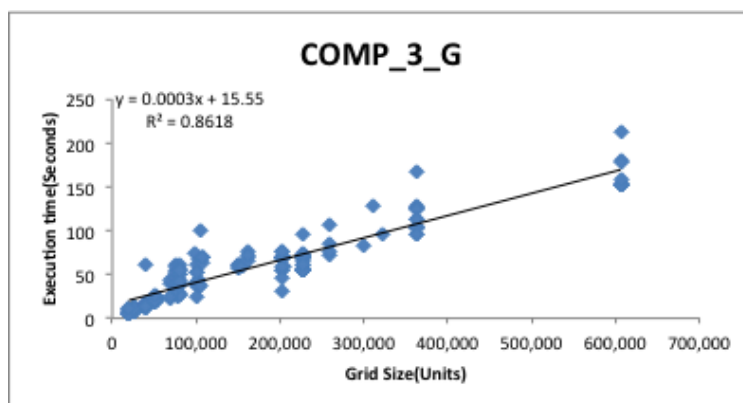


รูปที่ 5-17 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน ETACH ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ

ส่วนการทำงาน COMP\_3 (MAX, ETAMAP, CHANGE) สำหรับระดับความละเอียด R2 R3 R4 ที่ใช้เวลาในการทำงานขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล ซึ่งจะแสดงทั้งชื่อส่วนการทำงานที่ตามด้วย \_C คือใช้หน่วยประมวลผลกลางในการคำนวณ และ \_G คือใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ ตามรูปที่ 5-18 ถึงรูปที่ 5-19 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล ซึ่งจะกำหนดตัวแปรสำหรับใช้ในสมการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $Ap$  คือขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล

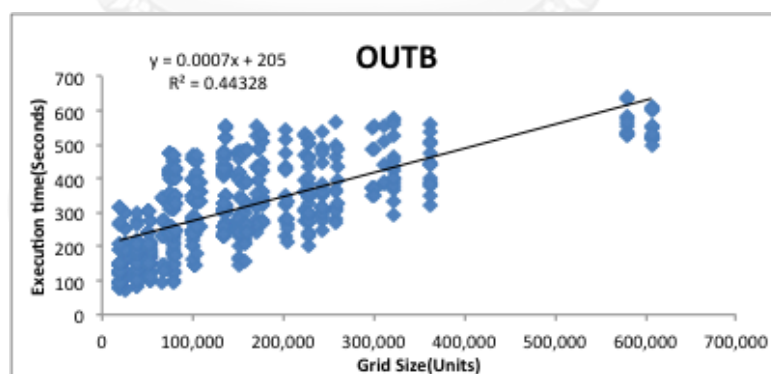


รูปที่ 5-18 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP\_3 ที่ใช้หน่วยประมวลผลกลางในการคำนวณ

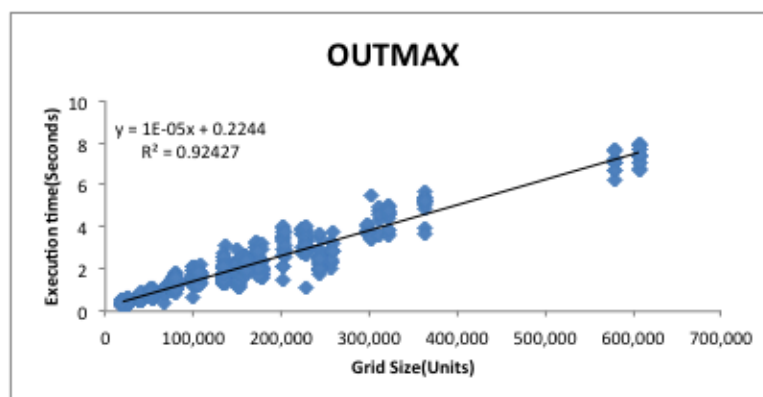


รูปที่ 5-19 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP\_3 ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ

ส่วนการทำงาน OUTB และ OUTMAX สำหรับทุกระดับความละเอียดที่ใช้เวลาในการทำงานขึ้นกับขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล ตามรูปที่ 5-20 ถึงรูปที่ 5-21 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล ซึ่งจะกำหนดตัวแปรสำหรับใช้ในการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $A$  คือขนาดของปัญหาทั้งหมดในแต่ละหน่วยประมวลผล



รูปที่ 5-20 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน OUTB

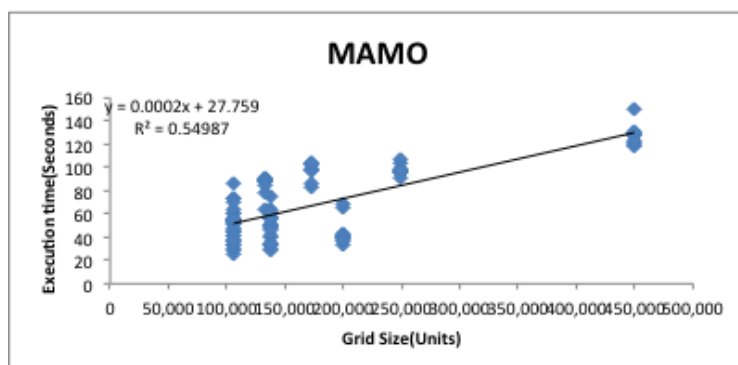


รูปที่ 5-21 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน OUTMAX

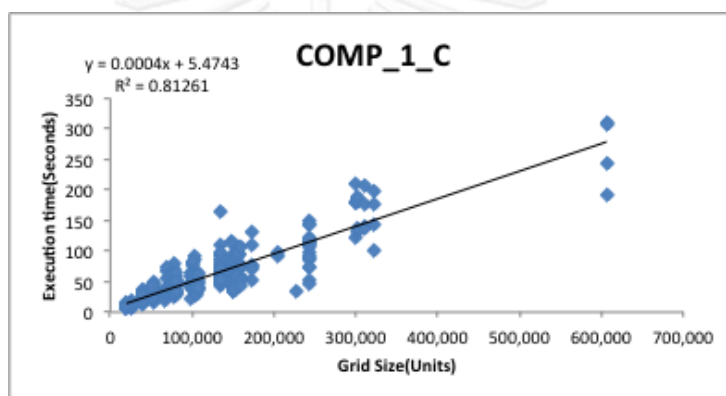
### 5.2.1.3 ส่วนการทำงานที่เวลาขึ้นกับขนาดของปัญหาเฉพาะพื้นที่น้ำในแต่ละหน่วย

#### ประมวลผล

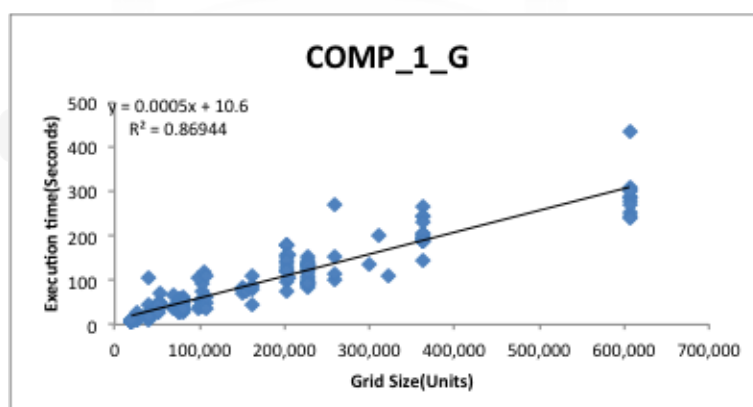
ส่วนการทำงาน MAMO (preMASS, postMASS, preMOMENT, postMOMENT, AFDEFORM, OPENBOUND) สำหรับระดับความละเอียด R1 COMP\_1 (preNLMASS, postNLMASS) และ COMP\_2 (preNLMMT1, postNLMMT1, preNLMMT2, postNLMMT2) สำหรับระดับความละเอียด R2 R3 R4 ที่ใช้เวลาในการทำงานขึ้นกับขนาดของปัญหาเฉพาะพื้นที่น้ำในแต่ละหน่วยประมวลผล ซึ่งจะแสดงทั้งชื่อส่วนการทำงานที่ตามด้วย \_C คือใช้หน่วยประมวลผลกลางในการคำนวณ และ \_G คือใช้หน่วยประมวลผลกราฟิกในการคำนวณ ตามรูปที่ 5-22 ถึงรูปที่ 5-26 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของปัญหาเฉพาะพื้นที่น้ำในแต่ละหน่วยประมวลผล ซึ่งจะกำหนดตัวแปรสำหรับใช้ในสมการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $Aps$  คือขนาดของปัญหาเฉพาะพื้นที่น้ำในแต่ละหน่วยประมวลผล



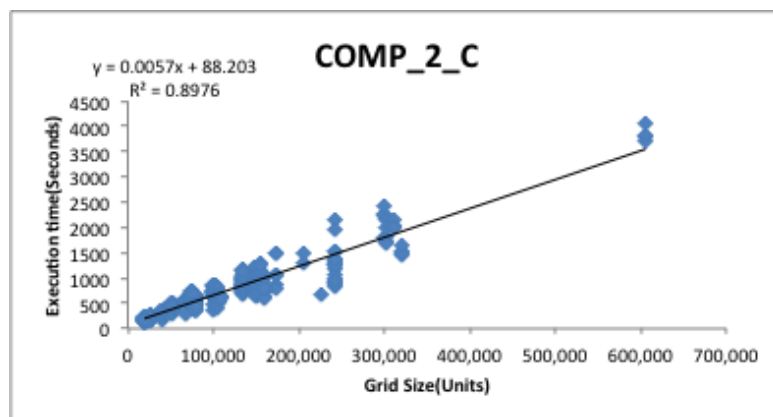
รูปที่ 5-22 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน MAMO



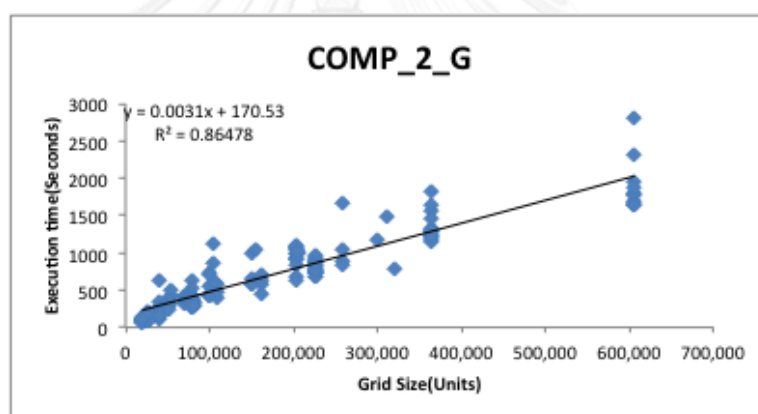
รูปที่ 5-23 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP\_1 ที่ใช้หน่วยประมวลผลกลางในการคำนวณ



รูปที่ 5-24 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP\_1 ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ



รูปที่ 5-25 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP\_2 ที่ใช้หน่วยประมวลผลกลางในการคำนวณ

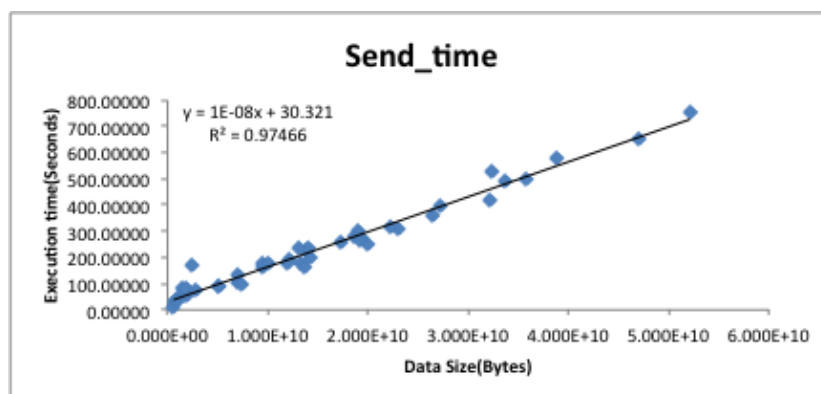


รูปที่ 5-26 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของปัญหาของส่วนการทำงาน COMP\_2 ที่ใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ

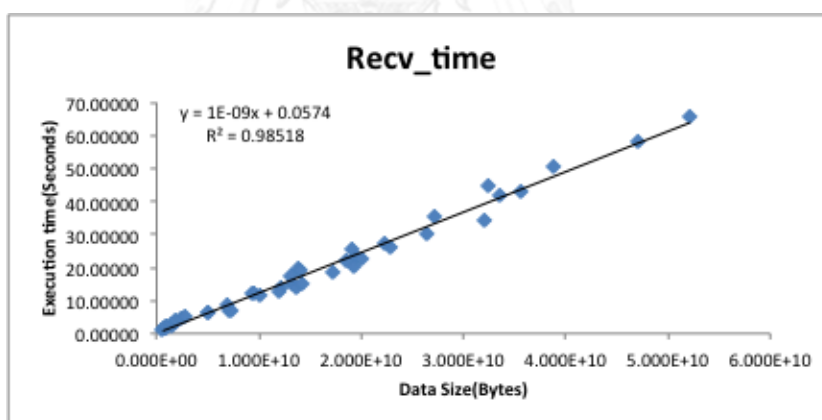
#### 5.2.1.4 ส่วนการทำงานที่ขึ้นกับปริมาณข้อมูลของการติดต่อสื่อสารระหว่างหน่วยประมวลผลในแต่ละครั้ง

ส่วนการทำงานที่เกี่ยวข้องกับการติดต่อสื่อสารระหว่างหน่วยประมวลผล อันได้แก่ Send (MPI\_Send) และ Recv (MPI\_Recv) สำหรับทุกระดับความละเอียดที่เวลาที่ใช้ในการทำงานขึ้นกับปริมาณของข้อมูลที่ใช้ในการสื่อสาร ตามรูปที่ 5-27 ถึงรูปที่ 5-28 โดยที่แกนตั้งแสดงถึงเวลาที่ใช้ในการทำงาน และแกนนอนแสดงถึงขนาดของปัญหาที่ใช้ในการทำงาน โดยมีตัวแปร  $y$  คือเวลาที่ใช้ในการทำงาน และ  $x$  คือขนาดของข้อมูลที่ใช้ในการสื่อสาร ซึ่งจะกำหนดตัวแปรสำหรับใช้ในสมการเชิงเส้นใหม่ได้แก่  $T$  คือเวลาที่ใช้ในการทำงาน และ  $B$  คือขนาดของข้อมูลที่ใช้ในการสื่อสาร ซึ่งคำนวณได้

จากขนาดของข้อมูลที่ใช้ในการติดต่อสื่อสาร (Length) คูณกับขนาดของตัวแปร (Variable size, 4 Bytes) และคูณกับจำนวนครั้งที่ต้องรับส่งข้อมูล (Timestep)



รูปที่ 5-27 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของข้อมูลที่ใช้ในการสื่อสารของส่วนการทำงาน Send



รูปที่ 5-28 ความสัมพันธ์ระหว่างเวลาที่ใช้งานกับขนาดของข้อมูลที่ใช้ในการสื่อสารของส่วนการทำงาน Recv

ส่วนเรื่องของการทำงานเกี่ยวกับการรอรับข้อมูลอันได้แก่ Wait (MPI\_wait) และ Barrier (S\_Barrier, W\_Barrier, E\_Barrier) สำหรับทุกระดับความละเอียด สามารถหาเวลาที่ใช้ได้จากการคำนวณเวลาที่มากที่สุดของการทำงานในแต่ละช่วงที่มีการรอกันระหว่างหน่วยประมวลผล

### 5.3 สรุปผลการวิเคราะห์

#### 5.3.1 สรุปผลการวิเคราะห์การทำงานของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม

ในการวิเคราะห์การทำงานของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานและแบบขนานลูกผสม สามารถสรุปความสัมพันธ์ของการใช้เวลาในการทำงานต่างๆออกมาเป็นสมการเชิงเส้น ( $Y = aX + b$ ) ดังตารางที่ 5-8

ตารางที่ 5-8 ค่าตัวแปรในสมการเชิงเส้นของส่วนการทำงานของโปรแกรมจำลองสึนามิแบบขนานลูกผสม

Region	Function	a	b	R <sup>2</sup>	Y	X
R1 R2 R3 R4	Initial	7.00E-07	1.046	0.07659	T	A
R1	MAMO	2.00E-04	27.759	0.550	T	Aps
R1	INTERQT_C	2.00E-04	104.740	0.512	T	Ap
R1	MAX	3.00E-04	24.407	0.780	T	Ap
R1	ETACH_C	3.00E-05	4.540	0.71094	T	Ap
R2	JNQ_S2C	1.00E-05	-0.645	0.74939	T	A
R3 R4	JNZ	1.00E-05	0.148	0.56087	T	A
R3 R4	JNQ	5.00E-06	0.324	0.5547	T	A
R2 R3 R4	Comp_1_C	4.00E-04	5.474	0.81261	T	Aps
R2 R3 R4	Comp_2_C	5.70E-03	88.203	0.8976	T	Aps
R2 R3 R4	Comp_3_C	3.00E-04	7.057	0.74694	T	Ap
R1 R2 R3 R4	OUTB	7.00E-04	205.030	0.44381	T	Ap
R1 R2 R3 R4	OUTMAX	1.00E-05	0.230	0.92561	T	Ap
R1 R2 R3 R4	Send	1.00E-08	30.321	0.9646	T	B
R1 R2 R3 R4	Recv	1.00E-09	0.057	0.9917	T	B
R1	INTERQT_G	2.00E-04	81.793	0.644	T	Ap
R1	ETACH_G	3.00E-05	6.270	0.7333	T	Ap
R2 R3 R4	Comp_1_G	5.00E-04	10.600	0.86944	T	Aps



R2 R3 R4	Comp_2_G	3.10E-03	170.530	0.86478	T	Aps
R2 R3 R4	Comp_3_G	3.00E-04	15.550	0.8618	T	Ap

เมื่อทำการแทนค่าตัวแปรในสมการเชิงเส้นของส่วนการทำงานของโปรแกรมจำลองสึนามิ แบบขนานลูกผสมที่หาได้ ตามระดับความละเอียดต่างๆของปัญหา จะได้ผลของการประมาณค่าเวลาที่ใช้ในการทำงานของส่วนการทำงานต่างๆดังตารางที่ 5-9 ซึ่งจะทำให้การหาความสัมพันธ์ของเวลาที่ใช้ในแต่ละระดับความละเอียดออกมาเป็นอัตราส่วนการทำงานที่แตกต่างกันในแต่ละระดับความละเอียด

ตารางที่ 5-9 ผลการประมาณเวลาที่ใช้ในการทำงานของแต่ละระดับความละเอียดของโปรแกรม  
จำลองสึนามิที่คำนวณแบบขนานลูกผสม

Region	X	A	s	Initial	JNZ	JNQ	Comp_1	Comp_2	Comp_3	OUTB	OUTMAX	Total
1	690	579,600	0.7758	1.452			141.352	197.713	198.287	610.750	6.026	1,155.580
21	721	606,361	0.9022	1.471			284.123	1,866.370	197.458	629.483	6.294	2,990.615
22	721	606,361	0.8191	1.471			258.934	1,710.198	197.458	629.483	6.294	2,809.255
23	721	606,361	0.4938	1.471			160.298	1,098.658	197.458	629.483	6.294	2,099.079
211	241	78,325	0.8633	1.101	0.931	0.715	44.410	380.152	39.048	259.858	1.013	727.227
212	637	298,753	0.9221	1.255	3.135	1.817	148.338	1,024.503	105.176	414.157	3.217	1,701.598
213	397	83,767	0.7279	1.105	0.985	0.742	41.089	359.559	40.680	263.667	1.068	708.895
214	241	55,189	0.7567	1.085	0.700	0.600	31.480	299.983	32.107	243.662	0.782	610.397
215	469	326,893	0.5740	1.275	3.417	1.958	104.418	752.202	113.618	433.855	3.499	1,414.241
221	639	311,193	0.7542	1.264	3.260	1.880	127.958	898.150	108.908	422.865	3.342	1,567.626
231	655	305,230	0.2978	1.260	3.200	1.850	56.056	452.357	107.119	418.691	3.282	1,043.815
232	637	296,842	0.7094	1.254	3.116	1.808	115.887	823.309	104.603	412.819	3.198	1,465.995
2111	487	158,275	0.8320	1.157	1.731	1.115	76.446	578.775	63.033	315.823	1.813	1,039.891
2112	487	205,027	0.8737	1.190	2.198	1.349	100.162	725.811	77.058	348.549	2.280	1,258.596
2121	643	301,567	0.8242	1.257	3.163	1.831	134.881	941.072	106.020	416.127	3.246	1,607.598
2122	679	505,855	0.8700	1.400	5.206	2.853	230.658	1,534.887	167.307	559.129	5.288	2,506.727

2131	595	357,595	0.6888	1.297	3.724	2.112	133.755	934.091	122.829	455.347	3.806	1,656.959
2141	379	241,423	0.6385	1.215	2.562	1.531	87.674	648.389	87.977	374.026	2.644	1,206.018
2151	271	221,407	0.5675	1.201	2.362	1.431	73.426	560.051	81.972	360.015	2.444	1,082.902
2152	739	364,327	0.4179	1.301	3.791	2.145	86.719	642.465	124.848	460.059	3.873	1,325.201
2153	523	276,667	0.4739	1.240	2.914	1.707	76.161	577.005	98.550	398.697	2.997	1,159.270
2211	703	321,271	0.5683	1.271	3.361	1.930	101.893	736.544	111.931	429.920	3.443	1,390.291
2212	703	346,579	0.5485	1.289	3.614	2.056	105.647	759.821	119.524	447.635	3.696	1,443.282
2213	703	295,963	0.4936	1.253	3.107	1.803	83.639	623.369	104.339	412.204	3.190	1,232.904
2311	667	304,819	0.3536	1.260	3.196	1.848	64.490	504.645	106.996	418.403	3.278	1,104.115
2312	631	288,367	0.4887	1.248	3.031	1.765	81.068	607.429	102.060	406.887	3.114	1,206.602
2313	703	295,963	0.3491	1.253	3.107	1.803	62.266	490.859	104.339	412.204	3.190	1,079.022
2321	703	321,271	0.7775	1.271	3.361	1.930	135.496	944.885	111.931	429.920	3.443	1,632.236
2322	631	220,219	0.4878	1.200	2.350	1.425	64.317	503.572	81.616	359.183	2.432	1,016.095
2323	631	311,083	0.8037	1.264	3.259	1.879	135.607	945.570	108.875	422.788	3.341	1,622.582

เนื่องจากภาระงานที่แตกต่างของระดับความละเอียด R1 เมื่อเทียบกับระดับความละเอียด R2 R3 R4 จำเป็นจึงต้องหาอัตราส่วนของการทำงานที่ระดับ R1 เทียบกับระดับอื่นๆ ด้วยการเปรียบเทียบเวลาที่ใช้งานต่อจุดในแต่ละระดับความละเอียดดังตารางที่ 5-10 ที่แสดงผลการเปรียบเทียบเวลาที่ใช้ในการคำนวณปัญหาส่วนของพื้นน้ำ ซึ่งผลที่ได้จากการทดลองคือ ระดับความละเอียด R1 ใช้เวลาในการทำงานเป็น 1/3 ของระดับความละเอียดอื่นๆโดยเฉลี่ย

ตารางที่ 5-10 แสดงผลการเปรียบเทียบเวลาที่ใช้ในการคำนวณปัญหาส่วนของพื้นน้ำ

Region	A	As	Comp. time	Time / As	R1 : R234
1	579,600	449,654	1,155.580	0.002570	
21	606,361	547,059	2,990.615	0.005467	0.4701
22	606,361	496,670	2,809.255	0.005656	0.4544
23	606,361	299,421	2,099.079	0.007010	0.3666

211	78,325	67,618	727.227	0.010755	0.2390
212	298,753	275,480	1,701.598	0.006177	0.4161
213	83,767	60,974	708.895	0.011626	0.2210
214	55,189	41,762	610.397	0.014616	0.1758
215	326,893	187,637	1,414.241	0.007537	0.3410
221	311,193	234,702	1,567.626	0.006679	0.3848
231	305,230	90,897	1,043.815	0.011483	0.2238
232	296,842	210,580	1,465.995	0.006962	0.3692
2111	158,275	131,685	1,039.891	0.007897	0.3254
2112	205,027	179,132	1,258.596	0.007026	0.3658
2121	301,567	248,552	1,607.598	0.006468	0.3973
2122	505,855	440,094	2,506.727	0.005696	0.4512
2131	357,595	246,311	1,656.959	0.006727	0.3820
2141	241,423	154,149	1,206.018	0.007824	0.3285
2151	221,407	125,648	1,082.902	0.008619	0.2982
2152	364,327	152,252	1,325.201	0.008704	0.2953
2153	276,667	131,112	1,159.270	0.008842	0.2907
2211	321,271	182,578	1,390.291	0.007615	0.3375
2212	346,579	190,099	1,443.282	0.007592	0.3385
2213	295,963	146,087	1,232.904	0.008439	0.3045
2311	304,819	107,784	1,104.115	0.010244	0.2509
2312	288,367	140,925	1,206.602	0.008562	0.3002
2313	295,963	103,321	1,079.022	0.010443	0.2461
2321	321,271	249,788	1,632.236	0.006534	0.3933
2322	220,219	107,423	1,016.095	0.009459	0.2717
2323	311,083	250,017	1,622.582	0.006490	0.3960

### 5.3.2 สรุปผลการวิเคราะห์การสื่อสารของโปรแกรมจำลองสึนามิแบบขนานลูกผสม

การสื่อสารของโปรแกรมที่ได้ถูกจำแนกไว้สองส่วนคือ การสื่อสารภายในระดับความละเอียด และการสื่อสารระหว่างระดับความละเอียด ซึ่งใช้เวลาในการทำงานขึ้นกับปริมาณของข้อมูลที่ใช้ในการสื่อสาร ซึ่งสามารถคำนวณหาปริมาณข้อมูลที่ใช้ในการสื่อสาร (B) ได้ตามตารางที่ 5-11 และ ตารางที่ 5-12 ที่คำนวณค่า B เพื่อใช้สำหรับสมการเชิงเส้นของส่วนการทำงานติดต่อสื่อสาร

$$B = length * V_{size} * Timestep * Row$$

ตารางที่ 5-11 ค่าตัวแปรที่ใช้สำหรับการหาปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสารภายในระดับความละเอียดของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม

Intra	Variable	To	length	V_size	Timestep	Row
R1	ma	Up	lenX	4	9000	1
	na	Down	lenX	4	9000	1
	zb	Down	lenX	4	9000	1
R234	ma(x)	Up	lenX	4	36000	2
	na(x)	Down	lenX	4	36000	2
	dz(x)	Up	lenX	4	36000	2
	za(x)	Down	lenX	4	36000	1
	zb(x)	Down	lenX	4	36000	1
	dma(x)	Down	lenX	4	36000	1
	dna(x)	Down	lenX	4	36000	1

ตารางที่ 5-12 ค่าตัวแปรที่ใช้สำหรับการหาปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่างระดับความละเอียดของโปรแกรมจำลองสึนามิที่คำนวณแบบขนานลูกผสม

Inter	send	To	Length	V_size	Timestep	Row
R1	m_jnq(y)	R2	Yo+2	4	36000	2
	n_jnq(x)	R2	Xi+2	4	36000	2

R2	dz(y)	R3	Y <sub>0</sub>	4	36000	2
	dz(x)	R3	X <sub>i</sub>	4	36000	2
	mb(y)	R3	Y <sub>0+2</sub>	4	36000	2
	nb(x)	R3	X <sub>i+2</sub>	4	36000	2
R3	dz(y)	R4	Y <sub>0</sub>	4	36000	2
	dz(x)	R4	X <sub>i</sub>	4	36000	2
	dz(y)	R2	Y <sub>n</sub>	4	36000	2
	zb(x)	R2	X <sub>i</sub>	4	36000	2
	mb(y)	R4	Y <sub>0+2</sub>	4	36000	2
	nb(x)	R4	X <sub>i+2</sub>	4	36000	2
R4	dz(y)	R3	Y <sub>n</sub>	4	36000	2
	zb(x)	R3	X <sub>i</sub>	4	36000	2

การสื่อสารภายในระดับความละเอียดสามารถแบ่งออกได้เป็นสามส่วนคือ ส่วนบน (Top) ส่วนกลาง (Middle) และส่วนล่าง (Bottom) ที่มีปริมาณข้อมูลที่ใช้ในการรับส่ง (B) คงที่ตามขนาดของปัญหาในแนวนอน (x) และจำนวนครั้งในการรับส่งข้อมูล (S, R) จากตารางที่ 5-11 สามารถคำนวณหาเวลาที่ใช้ในการสื่อสารภายในระดับความละเอียดดังตารางที่ 5-13 โดยการนำไปใช้งานให้เลือกใช้ตามลักษณะของการแบ่งงานของแต่ละระดับความละเอียด

### จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 5-13 ปริมาณข้อมูล จำนวนครั้งในการรับส่งข้อมูล และเวลาที่ใช้ในการสื่อสารภายในระดับความละเอียดของโปรแกรม

Region	X	B	Intra_1				Bottom Comm (P=2)				Intra_2				Middle Comm (P>2)				Intra_1				Intra_2				TOP Comm (P=2)			
			S		R		Intra_1		Intra_2		S		R		Intra_1		Intra_2		S		R		Intra_1		Intra_2		S		R	
			S	R	S	R	Intra_1	Intra_2	S	R	S	R	Intra_1	Intra_2	S	R	S	R	Intra_1	Intra_2	S	R	Intra_1	Intra_2	S	R	Intra_1	Intra_2		
1	690	2.484E+07	1	1	0	1	30.385	0.058	2	2	1	1	60.770	30.385	1	1	1	0	30.385	30.327										
21	721	1.038E+08	2	2	2	4	60.770	60.886	4	4	6	6	121.540	182.310	2	2	4	2	60.770	121.424										
22	721	1.038E+08	2	2	2	4	60.770	60.886	4	4	6	6	121.540	182.310	2	2	4	2	60.770	121.424										

23	721	1.038E+08	2	2	2	4	60.770	60.886	4	4	6	6	121.540	182.310	2	2	4	2	60.770	121.424
211	241	3.470E+07	2	2	2	4	60.759	60.873	4	4	6	6	121.517	182.276	2	2	4	2	60.759	121.402
212	637	9.173E+07	2	2	2	4	60.763	60.879	4	4	6	6	121.527	182.290	2	2	4	2	60.763	121.411
213	397	5.717E+07	2	2	2	4	60.759	60.874	4	4	6	6	121.517	182.276	2	2	4	2	60.759	121.402
214	241	3.470E+07	2	2	2	4	60.758	60.873	4	4	6	6	121.516	182.274	2	2	4	2	60.758	121.401
215	469	6.754E+07	2	2	2	4	60.764	60.879	4	4	6	6	121.528	182.292	2	2	4	2	60.764	121.413
221	639	9.202E+07	2	2	2	4	60.764	60.879	4	4	6	6	121.527	182.291	2	2	4	2	60.764	121.412
231	655	9.432E+07	2	2	2	4	60.764	60.879	4	4	6	6	121.527	182.291	2	2	4	2	60.764	121.412
232	637	9.173E+07	2	2	2	4	60.763	60.879	4	4	6	6	121.527	182.290	2	2	4	2	60.763	121.411
2111	487	7.013E+07	2	2	2	4	60.760	60.875	4	4	6	6	121.521	182.281	2	2	4	2	60.760	121.405
2112	487	7.013E+07	2	2	2	4	60.761	60.877	4	4	6	6	121.523	182.284	2	2	4	2	60.761	121.407
2121	643	9.259E+07	2	2	2	4	60.763	60.879	4	4	6	6	121.527	182.290	2	2	4	2	60.763	121.411
2122	679	9.778E+07	2	2	2	4	60.768	60.884	4	4	6	6	121.536	182.304	2	2	4	2	60.768	121.420
2131	595	8.568E+07	2	2	2	4	60.765	60.880	4	4	6	6	121.529	182.294	2	2	4	2	60.765	121.414
2141	379	5.458E+07	2	2	2	4	60.762	60.877	4	4	6	6	121.524	182.286	2	2	4	2	60.762	121.409
2151	271	3.902E+07	2	2	2	4	60.762	60.877	4	4	6	6	121.523	182.285	2	2	4	2	60.762	121.408
2152	739	1.064E+08	2	2	2	4	60.765	60.880	4	4	6	6	121.530	182.294	2	2	4	2	60.765	121.414
2153	523	7.531E+07	2	2	2	4	60.763	60.878	4	4	6	6	121.526	182.289	2	2	4	2	60.763	121.410
2211	703	1.012E+08	2	2	2	4	60.764	60.879	4	4	6	6	121.528	182.292	2	2	4	2	60.764	121.412
2212	703	1.012E+08	2	2	2	4	60.764	60.880	4	4	6	6	121.529	182.293	2	2	4	2	60.764	121.413
2213	703	1.012E+08	2	2	2	4	60.763	60.879	4	4	6	6	121.527	182.290	2	2	4	2	60.763	121.411
2311	667	9.605E+07	2	2	2	4	60.764	60.879	4	4	6	6	121.527	182.291	2	2	4	2	60.764	121.412
2312	631	9.086E+07	2	2	2	4	60.763	60.879	4	4	6	6	121.526	182.289	2	2	4	2	60.763	121.411
2313	703	1.012E+08	2	2	2	4	60.763	60.879	4	4	6	6	121.527	182.290	2	2	4	2	60.763	121.411
2321	703	1.012E+08	2	2	2	4	60.764	60.879	4	4	6	6	121.528	182.292	2	2	4	2	60.764	121.412

2322	631	9.086E+07	2	2	2	4	60.762	60.877	4	4	6	6	121.523	182.285	2	2	4	2	60.762	121.408
2323	631	9.086E+07	2	2	2	4	60.764	60.879	4	4	6	6	121.527	182.291	2	2	4	2	60.764	121.412

การสื่อสารระหว่างระดับความละเอียดสามารถแบ่งออกได้เป็นสองส่วนคือ ส่วนการส่งข้อมูลจากระดับความละเอียดต่ำไปสูง (L>H) และส่วนการส่งข้อมูลจากระดับความละเอียดสูงไปต่ำ (H>L) ที่มีปริมาณข้อมูลที่ใช้ในการรับส่ง (B) ขึ้นกับลักษณะและจำนวนของการแบ่งงานในแต่ละระดับความละเอียดดังตารางที่ 5-14

ตารางที่ 5-14 ปริมาณข้อมูลที่ใช้ในการติดต่อสื่อสารระหว่างระดับความละเอียด

Region	Xi	Yi	Yo	Yn	o	Inter_1	inter_2	inter_3	inter_4
						B (L > H)	B (H > L)	B (L > H)	B (H > L)
21	90	105	105	105	1	5.731E+07	-	-	-
22	90	105	105	105	1	5.731E+07	-	-	-
23	90	105	105	105	1	5.731E+07	-	-	-
211	80	108	108	108	1	5.414E+07	-	5.530E+07	5.414E+07
212	212	156	156	156	1	1.060E+08	-	1.071E+08	1.060E+08
213	132	70	70	70	1	5.818E+07	-	5.933E+07	5.818E+07
214	80	76	76	76	1	4.493E+07	-	4.608E+07	4.493E+07
215	156	232	232	232	1	1.117E+08	-	1.129E+08	1.117E+08
221	213	162	162	162	1	1.080E+08	-	1.092E+08	1.080E+08
231	218	155	155	155	1	1.074E+08	-	1.086E+08	1.074E+08
232	212	155	155	155	1	1.057E+08	-	1.068E+08	1.057E+08
2111	162	108	108	108	1	7.776E+07	5.443E+07	7.891E+07	-
2112	162	140	140	140	1	8.698E+07	6.365E+07	8.813E+07	-
2121	214	156	156	156	1	1.066E+08	7.574E+07	1.077E+08	-
2122	226	248	248	248	1	1.365E+08	1.040E+08	1.377E+08	-

2131	198	200	200	200	1	1.146E+08	8.611E+07	1.158E+08	-
2141	126	212	212	212	1	9.734E+07	7.920E+07	9.850E+07	-
2151	90	272	272	272	1	1.043E+08	9.130E+07	1.054E+08	-
2152	246	164	164	164	1	1.181E+08	8.266E+07	1.192E+08	-
2153	174	176	176	176	1	1.008E+08	7.574E+07	1.020E+08	-
2211	234	152	152	152	1	1.112E+08	7.747E+07	1.123E+08	-
2212	234	164	164	164	1	1.146E+08	8.093E+07	1.158E+08	-
2213	234	140	140	140	1	1.077E+08	7.402E+07	1.089E+08	-
2311	222	152	152	152	1	1.077E+08	7.574E+07	1.089E+08	-
2312	210	152	152	152	1	1.043E+08	7.402E+07	1.054E+08	-
2313	234	140	140	140	1	1.077E+08	7.402E+07	1.089E+08	-
2321	234	152	152	152	1	1.112E+08	7.747E+07	1.123E+08	-
2322	210	116	116	116	1	9.389E+07	6.365E+07	9.504E+07	-
2323	210	164	164	164	1	1.077E+08	7.747E+07	1.089E+08	-

ตารางที่ 5-15 เวลาที่ใช้ในการติดต่อสื่อสารภายนอกระดับความละเอียด

Region	N	n	Inter_1		Inter_2		Inter_3		Inter_4	
			SendTO	RecvBY	SendTO	RecvBY	SendTO	RecvBY	SendTO	RecvBY
21	1	1	30.8941	0.1147	-	-	-	-	-	-
22	1	1	30.8941	0.1147	-	-	-	-	-	-
23	1	1	30.8941	0.1147	-	-	-	-	-	-
211	1	1	30.8624	0.1115	-	-	30.8740	0.1127	30.8624	0.1115
212	1	1	31.3808	0.1634	-	-	31.3924	0.1645	31.3808	0.1634
213	1	1	30.9028	0.1156	-	-	30.9143	0.1167	30.9028	0.1156
214	1	1	30.7703	0.1023	-	-	30.7818	0.1035	30.7703	0.1023
215	1	1	31.4384	0.1691	-	-	31.4500	0.1703	31.4384	0.1691
221	1	1	31.4010	0.1654	-	-	31.4125	0.1666	31.4010	0.1654



231	1	1	31.3952	0.1648	-	-	31.4068	0.1660	31.3952	0.1648
232	1	1	31.3780	0.1631	-	-	31.3895	0.1642	31.3780	0.1631
2111	1	1	31.0986	0.1352	30.8653	0.1118	31.1101	0.1363	-	-
2112	1	1	31.1908	0.1444	30.9575	0.1210	31.2023	0.1455	-	-
2121	1	1	31.3866	0.1640	31.0784	0.1331	31.3981	0.1651	-	-
2122	1	1	31.6861	0.1939	31.3607	0.1614	31.6976	0.1951	-	-
2131	1	1	31.4672	0.1720	31.1821	0.1435	31.4788	0.1732	-	-
2141	1	1	31.2944	0.1547	31.1130	0.1366	31.3060	0.1559	-	-
2151	1	1	31.3636	0.1617	31.2340	0.1487	31.3751	0.1628	-	-
2152	1	1	31.5018	0.1755	31.1476	0.1401	31.5133	0.1766	-	-
2153	1	1	31.3290	0.1582	31.0784	0.1331	31.3405	0.1594	-	-
2211	1	1	31.4327	0.1686	31.0957	0.1349	31.4442	0.1697	-	-
2212	1	1	31.4672	0.1720	31.1303	0.1383	31.4788	0.1732	-	-
2213	1	1	31.3981	0.1651	31.0612	0.1314	31.4096	0.1663	-	-
2311	1	1	31.3981	0.1651	31.0784	0.1331	31.4096	0.1663	-	-
2312	1	1	31.3636	0.1617	31.0612	0.1314	31.3751	0.1628	-	-
2313	1	1	31.3981	0.1651	31.0612	0.1314	31.4096	0.1663	-	-
2321	1	1	31.4327	0.1686	31.0957	0.1349	31.4442	0.1697	-	-
2322	1	1	31.2599	0.1513	30.9575	0.1210	31.2714	0.1524	-	-
2323	1	1	31.3981	0.1651	31.0957	0.1349	31.4096	0.1663	-	-

สำหรับการหาปริมาณข้อมูลที่ใช้ในการสื่อสารระหว่างระดับความละเอียด จำเป็นต้องทราบตัวแปรต่างๆที่สำคัญดังนี้ ขอบเขตตามแกนนอนและแกนตั้งที่อ้างอิงความยาวตามแกนของระดับที่มีความละเอียดต่ำกว่า ( $X_i, Y_i$ ) ขอบเขตตามแกนตั้งที่อ้างอิงความยาวจากการแบ่งงานบนระดับความละเอียดที่ต่ำกว่า ( $Y_0$ ) ขอบเขตตามแกนตั้งที่อ้างอิงความยาวจากการแบ่งงานบนระดับความละเอียดของระดับนั้นๆ ( $Y_n$ ) จำนวนหน่วยประมวลผลของระดับที่มีความละเอียดต่ำกว่า ( $N$ ) จำนวนหน่วยประมวลผลของระดับที่มีความละเอียดสูงกว่า ( $n$ ) และจำนวนขอบเขตของระดับความละเอียดที่สูง

กว่าที่ถูกแบ่งโดยจำนวนหน่วยประมวลผลของระดับที่มีความละเอียดต่ำกว่า (o) เพื่อใช้สำหรับ  
คำนวณหาเวลาที่ใช้ในการสื่อสารดังตารางที่ 5-15

### 5.3.3 ลำดับการทำงานของโปรแกรมแบบขนานลูกผสม

ตารางที่ 5-16 ลำดับของการทำงานที่มีการรอรับข้อมูลของโปรแกรมจำลองสัญญาณที่คำนวณแบบ  
ขนานลูกผสม

Region	T0	T1	T2	T3	T4	T5	T6	T7	T8	Total
R1	INITIAL	R1_1	R1_2	-	-	-	-	-	OUTMAX	MAXTIME
R2		W1	R2_1	W4	R2_2	R2_3	-	-		
R3		R3_1	W2	R3_2	W5	R3_3	W7	R3_4		
R4		R4_1	W3	R4_2	R4_3	W6	R4_4	-		

จากตารางที่ 5-16 แสดงลำดับการทำงานทั้งหมดของโปรแกรมซึ่งในแต่ละระดับความ  
ละเอียดมีการรอรับข้อมูลจากระดับอื่นในช่วงเวลาที่แตกต่างกัน ซึ่งจะมีรายละเอียดดังต่อไปนี้

- ส่วนการทำงาน INITIAL และ OUTMAX ของทุกระดับความละเอียด
- R1\_1 [ Comp\_1 Comp\_2 Intra\_1 Intra\_2 และ Inter\_1(Send) ]
- R1\_2 [ MAX และ OUTB ]
- R2\_1 [ JNQ\_S2C Comp\_1 Intra\_1 และ Inter\_1 ]
- R2\_2 [ Comp\_2 Intra\_2 Inter\_3(Recv) และ Inter\_4(Send) ]
- R2\_3 [ Comp\_3 และ OUTB ]
- R3\_1 [ Comp\_1 Intra\_1 และ Inter\_1(Send) ]
- R3\_2 [ JNZ Inter\_1(Recv) และ Inter\_3(send) ]
- R3\_3 [ Comp\_2 Intra\_2 Inter\_2(Recv) และ Inter\_3(Send) ]
- R3\_4 [ JNQ Comp\_3 และ OUTB ]
- R4\_1 [ Comp\_1 และ Intra\_1 ]
- R4\_2 [ JNZ Inter\_1(Recv) และ Inter\_2(send) ]
- R4\_3 [ Comp\_2 และ Intra\_2 ]



## บทที่ 6

### การประเมินผลการทดลอง

#### 6.1 การประเมินผล

ในการประเมินผลของการทำงานโปรแกรมจำลองสีนามิที่คำนวณแบบขนานลูกผสม ได้ทำการทดสอบตามแนวคิดในการประเมินผล เพื่อทราบถึงประสิทธิภาพจากการแบ่งงาน ความแม่นยำของผลลัพธ์ และเวลาที่คำนวณได้จากสมการเทียบกับเวลาที่ใช้คำนวณจริง ซึ่งมีรายละเอียดดังต่อไปนี้

##### 6.1.1 ข้อมูลที่ใช้ในการประเมินผล

ข้อมูลที่ใช้ในการประเมินผลเป็นข้อมูลชุดเดียวกับการทดลองในบทที่ 5 ข้อมูลจริงสำหรับการจำลองสีนามิบางส่วนดังตารางที่ 5-1

##### 6.1.2 เครื่องมือที่ใช้ในการประเมินผล

การประเมินผลงานวิจัยนี้จะใช้เครื่องจากระบบเพกาซัสคลัสเตอร์ ในการทดสอบการทำงานของโปรแกรมจำลองสีนามิที่คำนวณแบบขนานลูกผสม และนำผลจากการทำงานที่ได้มาประเมินในด้านต่างๆ

##### 6.1.3 รูปแบบของการประเมินผล

รูปแบบข้อมูลที่ใช้ในการประเมินผลสำหรับหัวข้อทั้งหมดมีรายละเอียดดังนี้

1. ทดสอบข้อมูลโซน 2111 ที่กำหนดจำนวนประมวลผล 4 หน่วย
2. ทดสอบข้อมูลโซน 2111 ที่กำหนดจำนวนประมวลผล 16 หน่วย
3. ทดสอบข้อมูลโซน 2112 ที่กำหนดจำนวนประมวลผล 16 หน่วย
4. ทดสอบข้อมูลโซน 2121 ที่กำหนดจำนวนประมวลผล 16 หน่วย
5. ทดสอบข้อมูลโซน 2211 ที่กำหนดจำนวนประมวลผล 16 หน่วย
6. ทดสอบข้อมูลหลายโซน 2111/2112/2121/2211 ที่กำหนดจำนวนประมวลผล 32

หน่วย

## 6.2 ผลการประเมิน

### 6.2.1 ประสิทธิภาพจากการแบ่งงาน

ในการประเมินผลประสิทธิภาพจากการแบ่งงานใช้การทดสอบข้อมูลโซน 2111 ที่กำหนดจำนวนหน่วยประมวลผล 16 หน่วย ในการทดสอบ ซึ่งผลของการคำนวณหาปริมาณหน่วยประมวลผลสำหรับการคำนวณในแต่ละระดับความละเอียดแสดงดังตารางที่ 6-2 โดยค่า Average Job Size ที่คำนวณได้คือ 42,679 ซึ่งการจัดเรียงข้อมูลเพื่อหาจำนวนของหน่วยประมวลผลที่ใช้งานจริงจะแสดงดังตารางที่ 6-1

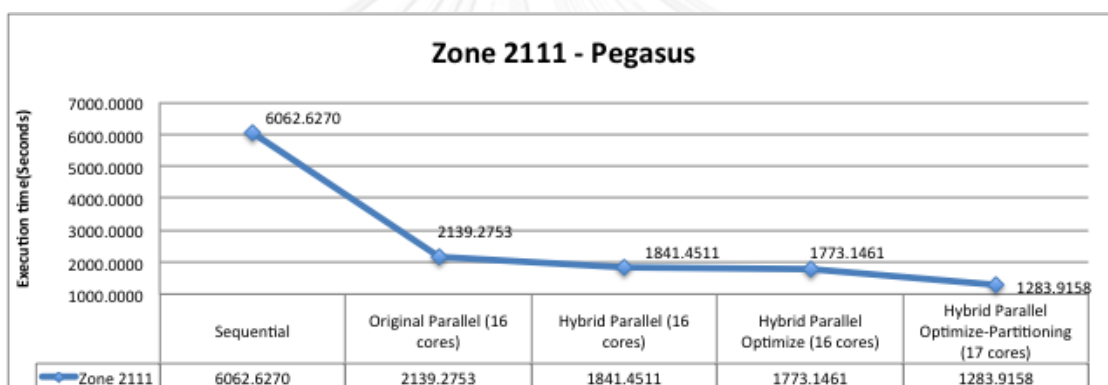
ตารางที่ 6-1 การจัดเรียงข้อมูลเพื่อหาจำนวนของหน่วยประมวลผลที่ใช้งานจริง

	p	2111	1.5	0.75	1											SUM	GPU
CPUS	3	R1	96,028	42,679	11,186											149,893	1
	10	R2	42,679	96,028	42,679	42,679	42,679	96,028	42,679	42,679	42,679	56,237				547,045	3
	2	R3	42,679	24,941												67,620	0
	2	R4	42,679	89,013												131,692	1
Total	17		0	1	2	3	4	5	6	7	8	9	10	11	12		

ตารางที่ 6-2 ตารางเปรียบเทียบการแบ่งงานในรูปแบบปกติซึ่งทุกระดับความละเอียดใช้จำนวนหน่วยประมวลผลในการคำนวณเท่ากัน (CUDA-Optimize) กับการเฉลี่ยงานตามประสิทธิภาพของหน่วยประมวลผล (CUDA-Optimize-Partitioning)

CPU=16	CUDA-Optimize	CUDA-Optimize-Partitioning
1	4	3
21	4	10
211	4	2
2111	4	2
SUM	16	17

โดยเวลาที่ใช้ในการคำนวณของรูปแบบต่างๆแสดงดังรูปที่ 6-1 ซึ่ง Sequential แสดงถึงโปรแกรมที่มีรูปแบบการคำนวณแบบลำดับ Original Parallel แสดงถึงโปรแกรมที่มีรูปแบบการคำนวณแบบขนาน Hybrid Parallel แสดงถึงโปรแกรมที่มีรูปแบบการคำนวณแบบขนานลูกผสมที่ไม่มีการกำหนดงานให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ Hybrid Parallel Optimize แสดงถึงโปรแกรมที่มีรูปแบบการคำนวณแบบขนานลูกผสมที่มีการกำหนดงานให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ Hybrid Parallel Optimize-Partitioning แสดงถึงโปรแกรมที่มีรูปแบบการคำนวณแบบขนานลูกผสมที่มีการกำหนดงานให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ และมีการเฉลี่ยงานตามประสิทธิภาพของหน่วยประมวลผล



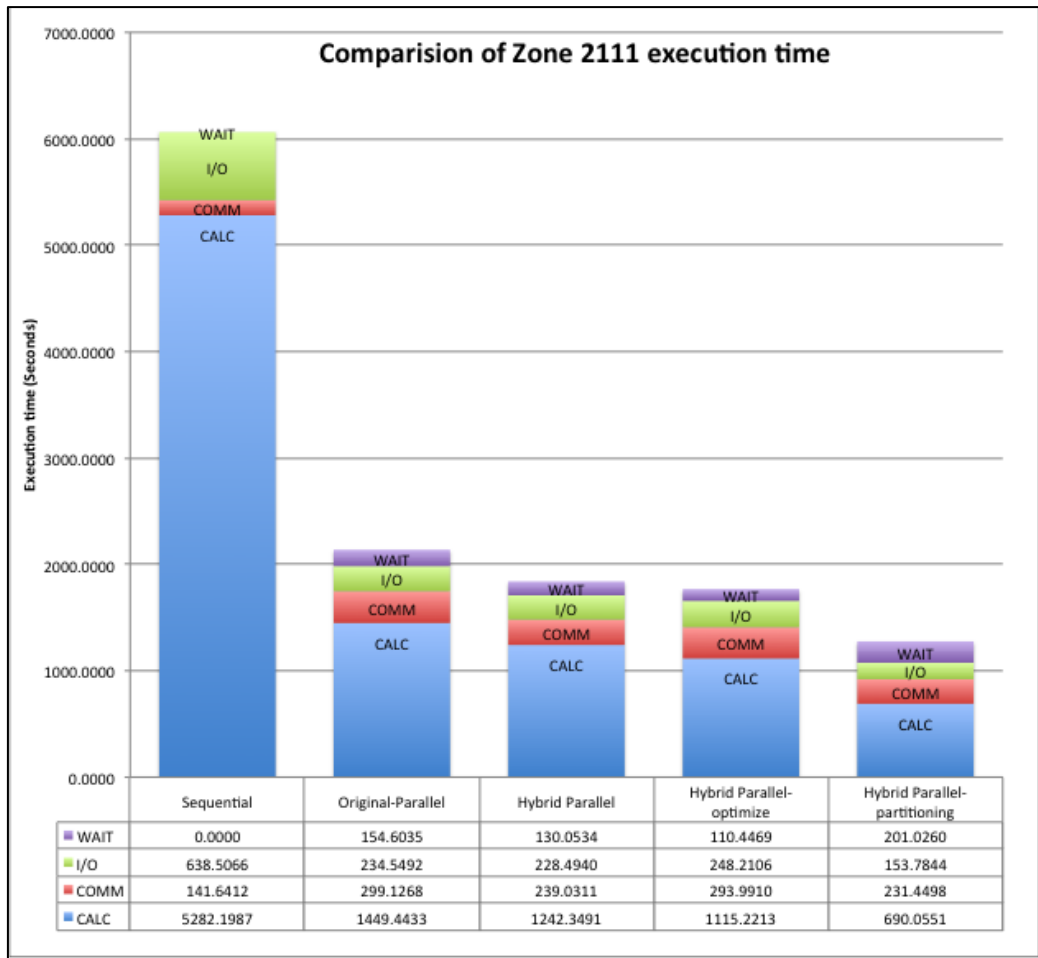
รูปที่ 6-1 แสดงเวลาที่ใช้ในการทำงานของโซน 2111 ของรูปแบบการทำงานต่างๆ

โดยจากรูปที่ 6-1 สามารถพิจารณาให้ละเอียดยิ่งขึ้นได้ตามรูปที่ 6-2 ซึ่งจะแสดงสัดส่วนของเวลาที่ใช้ในการทำงานต่างๆได้แก่ เวลาที่ใช้ในการคำนวณ (CALC) เวลาที่ใช้ในการสื่อสาร (COMM) เวลาที่ใช้ในการเขียนและอ่านไฟล์ (I/O) และเวลาที่ใช้ในการรอหน่วยประมวลผลอื่นเมื่อคำนวณเสร็จสิ้นแล้ว (WAIT) ซึ่งจะพบว่าเวลาที่ใช้ในการคำนวณจะลดลงอย่างชัดเจนเมื่อมีการปรับปรุงโปรแกรมให้สามารถติดต่อกับหน่วยประมวลผลกราฟิกส์เพื่อแบ่งงานไปคำนวณ และเวลาจะยิ่งลดลงมากขึ้นเมื่อได้กำหนดงานให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์และแบ่งงานอย่างเหมาะสมโดยการหาค่าปริมาณงานเฉลี่ย ซึ่งสามารถประเมินได้ว่าการแบ่งงานอย่างเหมาะสมสามารถเพิ่มประสิทธิภาพให้กับโปรแกรมที่คำนวณได้เป็นอย่างดี และผลความเร็วที่เพิ่มขึ้นของรูปแบบการทำงานทั้งหมดจากการปรับปรุงโปรแกรมจำลองสึนามิแสดงดังตารางที่ 6-3

เริ่มต้นจากโปรแกรมแบบลำดับ (Sequential program) มีสัดส่วนของการคำนวณเป็น 87.13% ของเวลาทั้งหมด โดยเวลาที่เหลือเป็นสื่อสารระหว่างหน่วยประมวลผลและการเขียนอ่านไฟล์ เมื่อนำมาปรับปรุงเป็น Original parallel program แล้วสามารถลดเวลาในการคำนวณได้ถึง 3.64 เท่า แต่เวลาทั้งหมดลดลงได้เพียง 2.83 เท่า เนื่องจากมีเวลาในการสื่อสารระหว่างหน่วยประมวลผลเพิ่มขึ้น (Communication time) และยังมี Wait time ซึ่งเป็นเวลาที่หน่วยประมวลผลต้องรอกันระหว่างไหมส์เต็ป แต่เวลา I/O ลดลงเนื่องจากการอ่านเขียนข้อมูลมีการซ้อนเหลื่อมกัน เมื่อนำมาปรับปรุงต่อเป็น Hybrid parallel โดยใช้ 4 GPU และ CPU 12 cores แล้วสามารถลดเวลาในการคำนวณได้อีก 17% เทียบกับ Original parallel แต่เวลาทั้งหมดลดลงได้ 16% โดยที่ยังมีเวลาที่ใช้ในการสื่อสารระหว่างหน่วยประมวลผล และเวลาในการอ่านเขียนไฟล์ใกล้เคียงเดิม จากนั้นเมื่อปรับปรุงต่อเป็น Hybrid parallel optimize โดยใช้ 4 GPU และ CPU 16 cores แล้วสามารถลดเวลาในการคำนวณได้อีก 11% เทียบกับ Hybrid parallel แต่เวลาทั้งหมดลดลงได้เพียง 4% สุดท้ายเมื่อจัดสรรงานอย่างเหมาะสมเป็น Hybrid parallel partitioning โดยใช้ 5 GPU และ CPU 17 cores แล้วสามารถลดเวลาในการคำนวณได้ถึง 62% เทียบกับ Hybrid parallel optimize แต่เวลาทั้งหมดลดลงได้เพียง 38% สรุปแล้วลดเวลาในส่วนการคำนวณได้ 7.65 เท่า และลดเวลาทั้งหมดได้ 4.72 เท่า เมื่อเทียบกับ Sequential program

ตารางที่ 6-3 ตารางแสดงความเร็วที่เพิ่มขึ้นจากการปรับปรุงโปรแกรมจำลองสึนามิ

Zone 2111	Speedup	
	CALC	Total
Original Parallel (4x4 CPU cores)	3.64	2.83
Hybrid Parallel (4 GPU + 4x3 CPU cores)	4.25	3.29
Hybrid Parallel Optimize (4 GPU + 4x4 CPU cores)	4.74	3.42
Hybrid Parallel Optimize-Partitioning (5 GPU + (4x4+1) CPU cores)	7.65	4.72



รูปที่ 6-2 รายละเอียดของเวลาที่ใช้ในการทำงานของโปรแกรมแบบขนานแบบต่างๆ

### 6.2.2 ความแม่นยำของผลลัพธ์

ในการประเมินผลประสิทธิภาพจากการแบ่งงานใช้การทดสอบข้อมูลโซน 2111 ที่กำหนดจำนวนหน่วยประมวลผล 4 หน่วย ในการทดสอบ ซึ่งการเปรียบเทียบผลลัพธ์ที่ได้จากการคำนวณเพื่อแสดงความแม่นยำของผลลัพธ์จากการใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณแสดงดังตารางที่ 6-4

ตารางที่ 6-4 การเปรียบเทียบผลลัพธ์ที่ได้จากการคำนวณเพื่อแสดงความแม่นยำของผลลัพธ์จากการใช้หน่วยประมวลผลกราฟิกส์ในการคำนวณ

Zone 2111	X	Y	ETA- fortran	ETA- cuda	diff.	Z4- fortran	Z4- cuda	diff.	M4- fortran	M4- cuda	diff.	N4- fortran	N4- cuda	diff.
ลิก 10	362	218	3696.000	3697.000	1.000	0.608	0.609	0.001	13.288	13.305	0.017	-0.217	-0.237	0.020



ม. KML														
ลี้ก 10 ม. PT	412	89	3982.000	3983.000	1.000	0.637	0.647	0.010	0.000	0.000	0.000	0.015	0.019	0.004
ริมหาด KML	392	218	4066.000	4066.000	0.000	0.446	0.447	0.001	11.889	11.890	0.001	-5.874	-5.828	0.046
ริมหาด PT	433	89	3848.000	3848.000	0.000	43.162	43.162	0.000	0.000	0.000	0.000	0.000	0.000	0.000

### 6.2.3 เวลาที่ใช้ในการคำนวณจริงกับเวลาที่คำนวณจากสูตรคำนวณเวลาที่ได้

ในการประเมินผลประสิทธิภาพจากการแบ่งงานใช้การทดสอบข้อมูลโซน 2111 2112 2121 และ 2211 ที่กำหนดจำหน่ายประมวลผล 16 หน่วย ในการทดสอบ ซึ่งผลของการประมาณเวลาในการทำงานของแต่ละโซนแสดงดังตารางที่ 6-5 ถึงตารางที่ 6-8

ตารางที่ 6-5 ตารางแสดงผลของการประมาณเวลาในการทำงานของโซน 2111

Region	T0	T1	T2	T3	T4	T5	T6	T7	T8	Total
1		571.714	374.337						1.679	
21	1.471	571.714	214.393	62.041	790.214	267.445				1908.957
211		149.977	636.130	62.041	0.000	458.933	331.282	261.458		
2111		93.239	56.737	63.259	427.596	666.249	274.400			

ตารางที่ 6-6 ตารางแสดงผลของการประมาณเวลาในการทำงานของโซน 2112

Region	T0	T1	T2	T3	T4	T5	T6	T7	T8	Total
1		571.714	374.337						1.679	
21	1.471	571.714	216.221	62.253	801.314	271.050				1925.701
211		149.977	637.958	62.253	0.000	458.933	342.381	261.558		
2111		93.239	56.737	63.259	427.596	668.289	274.400			

ตารางที่ 6-7 ตารางแสดงผลของการประมาณเวลาในการทำงานของโซน 2121

Region	T0	T1	T2	T3	T4	T5	T6	T7	T8	Total
1		467.199	422.637						2.162	1800.222
21	1.471	467.199	223.754	62.307	755.697	287.633				
211		149.977	540.976	62.307	0.000	458.933	296.764	261.282		
2111		93.239	56.737	63.259	427.596	571.361	274.400			

ตารางที่ 6-8 ตารางแสดงผลของการประมาณเวลาในการทำงานของโซน 2211

Region	T0	T1	T2	T3	T4	T5	T6	T7	T8	Total
1		528.987	422.637						2.162	1869.967
22	1.471	528.987	218.000	62.071	781.900	275.376				
221		180.807	566.179	62.071	0.000	489.759	292.141	261.383		
2211		153.997	26.810	63.155	488.469	566.385	274.400			

ซึ่งเมื่อนำข้อมูลสำหรับการทดสอบไปทำการประมวลผลจริง สามารถเก็บผลเวลาในการทำงานได้ดังตารางที่ 6-9 ซึ่งนำค่าประมาณเวลามาเปรียบเทียบกับเปอร์เซ็นต์ความคลาดเคลื่อนของการประมาณ

ตารางที่ 6-9 เวลาที่ใช้ทำงานจริงของข้อมูลทดสอบโดยระบบเพกาซัสคลัสเตอร์

Zone	Num. Proc.	Estimate time (Sec.)	Actual time (Sec.)	Different time (Sec.)	Error Percentage (%)
2111	17	1908.957	1283.916	625.041	32.743
2112	18	1925.701	1337.799	587.902	30.529
2121	17	1800.222	1733.857	66.365	3.686
2211	17	1869.967	1348.571	521.396	27.883

### 6.3 สรุปผลการประเมิน

ผลจากการประเมินจากการทดสอบพบว่า มีความคาดเคลื่อนสูงสุดอยู่ที่ 32.743% ซึ่งคาดว่า ความคาดเคลื่อนน่าจะมากจากการเก็บผลเวลาที่ใช้ในการทำงานของโปรแกรมจำลองสึนามิแบบขนานถูกผสม ซึ่งเวลาที่ทำได้ค่อนข้างที่จะไม่คงที่จึงส่งผลให้สมการที่ได้มีความคาดเคลื่อนค่อนข้างมาก

ตารางที่ 6-10 ตารางแสดงประสิทธิภาพและความเร็วที่เพิ่มขึ้นจากการปรับปรุงโปรแกรมจำลองสึนามิ

มิ

Zone	processors-ori_parallel	processors-h_parallel	Sequence	Original Parallel	Hybrid Parallel	Original Parallel Speedup	Hybrid Parallel Speedup	Original Parallel efficiency	Hybrid Parallel efficiency
2111	16	17	6062.6270	2139.2753	1283.9158	2.83	<b>4.72</b>	0.18	<b>0.28</b>
2112	16	18	6451.5960	2005.5204	1337.7986	3.22	<b>4.82</b>	0.20	<b>0.27</b>
2121	16	17	8517.574	2012.8645	1733.8573	4.23	<b>4.91</b>	0.26	<b>0.29</b>
2211	16	17	6841.4532	1936.6461	1348.5710	3.53	<b>5.07</b>	0.22	<b>0.30</b>

สำหรับประสิทธิภาพและความเร็วที่เพิ่มขึ้นของรูปแบบการทำงานทั้งหมดจากการปรับปรุงโปรแกรมจำลองสึนามิแสดงดังตารางที่ 6-10 พบว่าการปรับปรุงโปรแกรมจำลองสึนามิที่คำนวณแบบขนานให้สามารถคำนวณงานบนหน่วยประมวลผลกราฟิกส์ได้ด้วย ทำให้ความเร็วในการทำงานของโปรแกรมดีขึ้นอย่างมีนัยสำคัญ

## บทที่ 7

### สรุปผลการวิจัย

#### 7.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ได้นำเสนอการออกแบบและพัฒนาวิธีจัดการงานของโปรแกรมคำนวณแบบขนาน สำหรับปัญหาที่มีหลายระดับความละเอียดที่ทำงานบนคลัสเตอร์คอมพิวเตอร์ที่มีทั้งหน่วยประมวลผลแบบหลายคอร์และหน่วยประมวลผลกราฟิกส์ด้วยเทคนิคการเขียนโปรแกรมแบบลูกผสม ซึ่งสามารถปรับการจัดการงานให้เหมาะสมกับความสามารถของระบบคลัสเตอร์คอมพิวเตอร์ที่ใช้งานได้ โดยเทคนิคที่นำมาประยุกต์ใช้สำหรับการสื่อสารระหว่างหน่วยประมวลผลกลางกับหน่วยประมวลผลกราฟิกส์ได้แก่ “Zero-Copy Memory” และทำการวิเคราะห์การทำงานบนระบบคลัสเตอร์เพื่อได้เป็นส่วนส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกลางกับหน่วยประมวลผลกราฟิกส์ และทำการวิเคราะห์เพื่อได้ส่วนส่วนของงานที่เหมาะสมของหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ รวมไปถึงการจัดการขนาดของปัญหาให้เหมาะสมตามข้อกำหนดในเรื่องของหน่วยประมวลผลที่ใช้งาน

จากการทดลองในบทที่ 5 พบว่าส่วนส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์กับหน่วยประมวลผลกลางไม่สามารถกำหนดได้จากส่วนส่วนของประสิทธิภาพการทำงานของโปรแกรมแบบขนานลูกผสมกับโปรแกรมแบบขนาน แต่จะได้ส่วนที่เหมาะสมจากการทดลองเพิ่มเติมโดยค่าจากส่วนส่วนของประสิทธิภาพจะเป็นเพียงจุดตั้งต้นในการค้นหาส่วนที่เหมาะสมต่อไป การหาส่วนที่เหมาะสมของหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ก็เช่นเดียวกันต้องทำการทดลองเพื่อให้ได้ค่าที่เหมาะสมซึ่งผลของส่วนที่ได้ก็จะไม่เกิน 1.00 นั่นคือปริมาณของข้อมูลที่จะได้กับมากที่สุดก็จะได้รับงานเท่ากับหน่วยประมวลผลกลางตัวอื่นๆที่ไม่ได้ติดต่อกับหน่วยประมวลผลกราฟิกส์

จากการประเมินผลในบทที่ 6 พบว่าการแบ่งงานที่เหมาะสมด้วยวิธีเฉลี่ยงานสามารถช่วยลดเวลาในการคำนวณได้ดี ซึ่งจะสามารถแบ่งงานให้เหมาะสมได้จะต้องอาศัยค่า Average Job Size ซึ่งได้มาจากการคำนวณ ส่วนในเรื่องของความแม่นยำจากการใช้หน่วยประมวลผลกราฟิกส์นั้นพบว่าไม่มีปัญหาที่จะเพิ่มหน่วยประมวลผลดังกล่าวเข้ามาในระบบการทำงานเนื่องจากความผิดพลาดของผลลัพธ์มีค่าน้อยมาก แต่ในประเด็นจากการประเมินผลของสูตรคำนวณเวลาซึ่งผลลัพธ์ที่ได้จากการประมาณเวลานั้นยังมีความคลาดเคลื่อนที่สูงอยู่เมื่อเปรียบเทียบกับเวลาที่ทำได้จริง ซึ่งน่าจะเป็นผลมาจากความไม่คงที่ของเวลาในการประมวลผลแต่ละครั้งซึ่งได้นำไปวิเคราะห์เพื่อสร้างสมการ

โดยวิธีการจัดการงานสำหรับโปรแกรมคำนวณแบบขนานที่ปัญหาหลายระดับความละเอียดนี้ สามารถนำไปประยุกต์ใช้กับกรณีศึกษาอื่นๆหรือระบบคลัสเตอร์อื่นๆได้ โดยเพียงทำตามขั้นตอนที่สรุปมาได้ดังนี้ ก็จะได้รับประสิทธิภาพของการทำงานที่ดีขึ้นมากกว่าเดิมจากการได้รับสัดส่วนของงานที่เหมาะสม

1. ทำการพัฒนาโปรแกรมแบบขนานด้วยการเพิ่มส่วนที่ติดต่อกับหน่วยประมวลผลกราฟิกส์ โดยเน้นเลือกพัฒนาส่วนที่ใช้เวลาค่อนข้างมากเมื่อเปรียบเทียบกับเวลาที่ใช้ทั้งหมด พร้อมทั้งทำการทดสอบหาค่าสัดส่วนของงานที่เหมาะสมระหว่างหน่วยประมวลผลกราฟิกส์กับหน่วยประมวลผลกลาง และทำการทดสอบหาค่าสัดส่วนของงานที่สามารถกำหนดให้กับหน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์
2. ทำการเก็บข้อมูลเวลาของส่วนการทำงานในโปรแกรมแบบขนานลูกผสมที่ได้พัฒนา เพื่อสร้างสมการความสัมพันธ์เชิงเวลา โดยโปรแกรมที่ทดลองนั้นยังไม่ควรที่จะให้หน่วยประมวลผลกลางที่ติดต่อกับหน่วยประมวลผลกราฟิกส์นั้นได้รับงานไปคำนวณ
3. ทำการจัดสรรงานให้เหมาะสมโดยคำนวณหา Average Job Size เพื่อทราบปริมาณงานโดยเฉลี่ย แล้วจึงนำไปคำนวณหาหน่วยประมวลผลที่จะใช้จริงในแต่ละระดับความละเอียด จากนั้นต้องทำการเปลี่ยนงานให้เหมาะสมอีกครั้งให้พร้อมสำหรับการคำนวณจริง (Equivalent Average Job Size)
4. ทำการสร้างสูตรคำนวณเวลาที่ได้จากการเก็บข้อมูลของส่วนการทำงานทั้งหมดในโปรแกรม ซึ่งจำเป็นต้องวิเคราะห์รูปแบบการทำงานของโปรแกรมทั้งหมด เพื่อที่จะทราบถึงลักษณะการทำงานอย่างละเอียดและสามารถประเมินเวลาที่ใช้งานได้อย่างแม่นยำ

## 7.2 แนวทางในการวิจัยต่อ

1. นำวิธีการจัดการงานที่ได้ไปประยุกต์ใช้กับงานวิจัยในกรณีศึกษาอื่นๆ โดยอาจจะเพิ่มเทคนิคที่เกี่ยวกับหน่วยประมวลผลกราฟิกส์ให้มากขึ้น
2. ปรับปรุงวิธีการวิเคราะห์ข้อมูลเวลาให้สะดวกต่อการทำงานให้มากขึ้น โดยอาจจะเขียนเป็นโปรแกรมสำหรับวัดประสิทธิภาพในการทำงานของโปรแกรมออกมา พร้อมทั้งหาสัดส่วนที่เหมาะสมสำหรับงานแบบขนานลูกผสม
3. เนื่องจากในวิทยานิพนธ์นี้ได้นำเสนอวิธีการแบ่งงานให้เหมาะสมด้วยลักษณะคงที่ (static) ซึ่งเหมาะกับลักษณะการทำงานบนระบบคลัสเตอร์คอมพิวเตอร์แบบที่จัดสรรงานให้กับโหนดโดยไม่ซ้ำกัน จึงไม่มีการแบ่งทรัพยากรกันใช้ แต่หากนำไปใช้กับระบบคลัสเตอร์คอมพิวเตอร์แบบที่ยอมให้งานหลายงานทำงานบนโหนดเดียวกันได้ ก็อาจมีการแย่งใช้ทรัพยากรและจะ

ทำให้ลักษณะของการแบ่งงานไม่สอดคล้องกับประสิทธิภาพในการคำนวณของระบบในขณะนั้นซึ่งเป็นข้อจำกัดของวิทยานิพนธ์นี้ ซึ่งจะต้องมีการปรับปรุงกระบวนการแบ่งงานให้สามารถปรับเปลี่ยนสัดส่วนของภาระงานได้ตามสภาพแวดล้อมของการทำงาน (Dynamic scheduling) โดยอาจทำการแบ่งงานใหม่เป็นระยะๆ หรือเมื่อมีการตรวจพบความไม่สมดุลของภาระงาน (Load imbalance) ซึ่งสามารถพิจารณาได้จากเวลาที่แต่ละหน่วยประมวลผลทำงานเสร็จ



จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

## รายการอ้างอิง

1. Blaise Barney, L.L.N.L. *Introduction to Parallel Computing*. [cited 2013 1 April]; Available from: [https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/).
2. Wikipedia, t.f.e. *Multiscale modeling*. [cited 2013 1 April]; Available from: [http://en.wikipedia.org/wiki/Multiscale\\_modeling/](http://en.wikipedia.org/wiki/Multiscale_modeling/).
3. Wikipedia, t.f.e. *Simulation*. [cited 2013 1 April]; Available from: <http://en.wikipedia.org/wiki/Simulation/>.
4. Mathematics, D.o.C.a.A. *Multiscale in Biology*. [cited 2013 1 April]; Available from: [www.caam.rice.edu/~jy5/Biology2.ppt/](http://www.caam.rice.edu/~jy5/Biology2.ppt/).
5. Blaise Barney, L.L.N.L. *Message Passing Interface (MPI)*. [cited 2013 1 April]; Available from: <https://computing.llnl.gov/tutorials/mpi/>.
6. Corporation, N. *Parallel Program and Computing Platform | CUDA | NVIDIA*. [cited 2013 1 April]; Available from: [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html).
7. Ding, C. *CUDA Tutorial*. [cited 2013 1 April]; Available from: [http://geco.mines.edu/tesla/cuda\\_tutorial\\_mio/index.html/](http://geco.mines.edu/tesla/cuda_tutorial_mio/index.html/).
8. srl, E. *Off-line programming of robot and bending of process*. [cited 2013 1 April]; Available from: [http://www.robowave.com/index.php?option=com\\_content&view=article&id=61&Itemid=136&lang=en/](http://www.robowave.com/index.php?option=com_content&view=article&id=61&Itemid=136&lang=en/).
9. Hill, M.D. and M.R. Marty, *Amdahl's Law in the Multicore Era*. Computer, 2008. **41**(7): p. 33-38.
10. Jin, H. and R.F.V.d. Wijngaart, *Performance characteristics of the multi-zone NAS parallel benchmarks*. J. Parallel Distrib. Comput., 2006. **66**(5): p. 674-685.
11. Wikipedia, t.f.e. *Speedup*. [cited 2013 1 April]; Available from: <http://en.wikipedia.org/wiki/Speedup/>.
12. Wikipedia, t.f.e. *Efficiency*. [cited 2013 1 April]; Available from: <http://en.wikipedia.org/wiki/Efficiency/>.
13. Sourceforge. *IPM – Overview*. [cited 2013 1 April]; Available from: <http://ipm-hpc.sourceforge.net/>.
14. Sourceforge. *Ploticus: welcome*. [cited 2013 1 April]; Available from: <http://ploticus.sourceforge.net/>.
15. สิทธิกร ถาวรรัตนวิช, การพัฒนาโปรแกรมคำนวณแบบขนานที่ปรับปรุงประสิทธิภาพได้สำหรับปัญหาที่หลายระดับความละเอียด) กรณีศึกษาการจำลองสึนามิ: *Parallel Program Development with Adaptive Performance Tuning for Multi-scale Problem:A*

- case study of Tsunami Simulation*), 2553, จุฬาลงกรณ์มหาวิทยาลัย: วิศวกรรมคอมพิวเตอร์ วิศวกรรมศาสตร์.
16. Sourceforge. *mpiP: Lightweight, Scalable MPI Profiling*. [cited 2013 1 April]; Available from: <http://mpip.sourceforge.net/>.
  17. เขม อำนวยลาภ, ฉัตรชัย พีรพัฒน์ดิษฐ์, ระบบเตือนภัยสึนามิด้วยการจำลองแบบทันกาลบนหน่วยประมวลผลกราฟิกส์ )*Tsunami Warning System with Real-Time Simulation on Graphics Processing Unit*), 2553, จุฬาลงกรณ์มหาวิทยาลัย: วิศวกรรมคอมพิวเตอร์ วิศวกรรมศาสตร์.
  18. Gidra, H., et al., *Parallelizing TUNAMI-N1 Using GPGPU*, in *Proceedings of the 2011 IEEE International Conference on High Performance Computing and Communications 2011*, IEEE Computer Society. p. 845-850.
  19. Playne, D.P. and K.A. Hawick, *Asynchronous Communication Schemes for Finite Difference Methods on Multiple GPUs*, in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing 2010*, IEEE Computer Society. p. 763-768.
  20. Dr.Fumihiko Imamura, D.A.C.Y., and Res. Assist. Gulizar Ozyurt. *Tsunami Modeling Manual (TUNAMI model)*. 2006 [cited 2013 1 April]; Available from: <http://www.tsunami.civil.tohoku.ac.jp/hokusai3/J/projects/manual-ver-3.1.pdf>.
  21. กิตติพัฒน์ วิโรจน์ศิริ, โปรแกรมแบบขนานเพื่อการจำลองสึนามิ )*Parallel Tsunami Simulation Program*), 2550, จุฬาลงกรณ์มหาวิทยาลัย: วิศวกรรมคอมพิวเตอร์ วิศวกรรมศาสตร์.





ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

## ภาคผนวก ก

### Configuration file ของโปรแกรม

#### ตัวแปรที่ใช้ในโปรแกรม

ตัวแปรที่ใช้ในการคำนวณโดยหลักประกอบด้วย

Z – การกระจัดของผิวน้ำในแกน Z

M – คลื่นในแนวแกน X

N – คลื่นในแนวแกน Y

DZ – ความลึกของน้ำ

DM – ความลึกของน้ำบริเวณหน้าคลื่นในแนวแกน X

DN – ความลึกของน้ำบริเวณหน้าคลื่นในแนวแกน Y

โดยที่ a หลังตัวแปรต่าง ๆ แทนค่าของตัวแปรนั้น ณ เวลา และ 1-tb หลังตัวแปรต่าง ๆ แทนค่าของตัวแปรนั้น ณ เวลา t

#### การเตรียมแฟ้มโครงสร้างข้อมูลของโปรแกรม

แฟ้มโครงสร้างข้อมูลของโปรแกรม (Data configuration file) เป็นข้อมูลที่ใช้ในการตั้งค่าตัวแปรต่าง ๆ ที่ใช้ในการคำนวณของโปรแกรม เพื่อให้สะดวกในการปรับเปลี่ยนข้อมูลสำหรับปัญหาที่มีหลายกรณีในการคำนวณ แทนการกำหนดค่าลงในโปรแกรมโดยตรง ซึ่งในงานวิจัยนี้ได้ใช้แฟ้มข้อมูลด้วยรูปแบบโครงสร้าง XML (eXtensible Markup Language File) หรือรูปแบบของรากต้นไม้ มาใช้ในการกำหนดค่าตัวแปรต่าง ๆ ของปัญหาในแต่ละระดับความละเอียดสำหรับโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนาน ที่บรรจุตัวแปรต่าง ๆ ที่มีในโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ ซึ่งมีโครงสร้างหลักดังนี้

- <PTUNAMI>...</PTUNAMI> ข้อมูลทั่วไปทั้งหมดต้องถูกระบุภายใน นี้ tag

- <REGION1>...</REGION1> ข้อมูลของ R1 ต้องถูกระบุใน นี้ tag ซึ่งอยู่ภายใต้ <PTUNAMI>...</PTUNAMI>
- <REGION2>...</REGION2> ข้อมูลของ R2 ต้องถูกระบุใน นี้ tag ซึ่งอยู่ภายใต้ <REGION1>...</REGION1>
- <REGION3>...</REGION3> ข้อมูลของ R3 ต้องถูกระบุใน นี้ tag ซึ่งอยู่ภายใต้ <REGION2>...</REGION2>
- <REGION4>...</REGION4> ข้อมูลของ R4 ต้องถูกระบุใน นี้ tag ซึ่งอยู่ภายใต้ <REGION3>...</REGION3>

```

<?xml version="1.0" encoding="UTF-8"?>
<PTUNAMI>
...
    <REGION1>
    ...
        <REGION2>
        ...
            <REGION3>
            ...
                <REGION4>
                ...
                    </REGION4>
                </REGION3>
            </REGION2>
        </REGION1>
    </PTUNAMI>

```

รูปที่ ก-1 ตัวอย่างโครงสร้าง Configuration file

ส่วนที่ใช้ในการระบุการแบ่งพื้นที่การคำนวณในแนวตั้งและแนวนอนของแต่ละระดับที่มีอยู่ใน tag ของ REGION1 REGION2 REGION3 และ REGION4 คือ

```
<PCOL value="1" />
```

```
<PROW value="1" />
```

สำหรับ Configuration file ที่ใช้ระบุของพื้นที่ในการคำนวณที่มีมากกว่าหนึ่งโซน จะมีโครงสร้างของ ดัง Configuration file รูปที่ ก-2

```
<PTUNAMI>
...
  <REGION1>
  ...
    <REGION2>
    ...
      <REGION3>
      ...
        <REGION4>
        ...
          </REGION4>
          <REGION4>
          ...
          </REGION4>
        </REGION3>
      <REGION3>
      ...
        <REGION4>
        ...
        </REGION4>
      </REGION3>
    </REGION2>
  <REGION2>
  ...
    <REGION3>
    ...
      <REGION4>
      ...
      </REGION4>
    </REGION3>
  </REGION2>
</REGION1>
</PTUNAMI>
```

รูปที่ ก-2 โครงสร้าง Configuration file สำหรับข้อมูลพื้นที่มากกว่าหนึ่งโซน

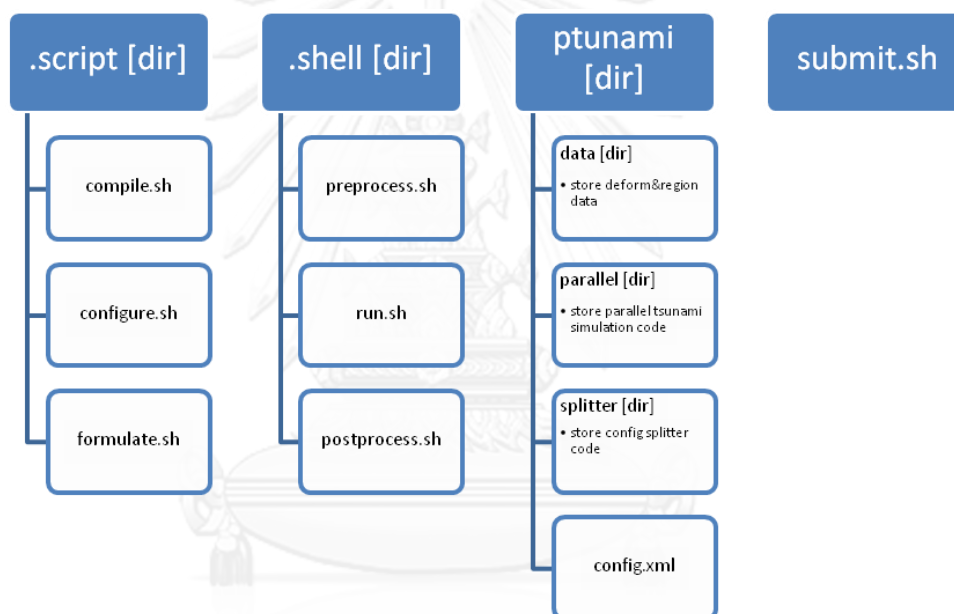
## ภาคผนวก ข

### โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนาน

#### Parallel Tsunami Simulation Manual

#### ส่วนประกอบของไฟล์

โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานประกอบด้วยไฟล์ข้อมูลต่าง ๆ ที่ใช้ในการทำงาน ดัง



รูปที่ ข-1 โครงสร้างของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนาน  
การใช้งานโปรแกรม

โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานสามารถสั่งงานได้ด้วยคำสั่ง

`./submit.sh [Cluster name] [Task name] [Config file] [TimeofJob]`

- [Cluster name] – ซึ่งในงานวิจัยนี้จะต้องเลือก rocks
- [Task name] – ชื่องานสำหรับการส่งงานเข้าคิวบนคลัสเตอร์
- [Config file] – ตำแหน่งและชื่อของ Configuration file
- [TimeofJob] – กำหนดระยะเวลาสำหรับงานที่รัน โดยขึ้นกับ Configuration ของ Queue ในระบบคลัสเตอร์

### ประวัติผู้เขียนวิทยานิพนธ์

นายธารนธิ เศรษฐพานิชผล เกิดเมื่อวันที่ 24 ตุลาคม พ.ศ. 2532 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2553 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2555



จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY