

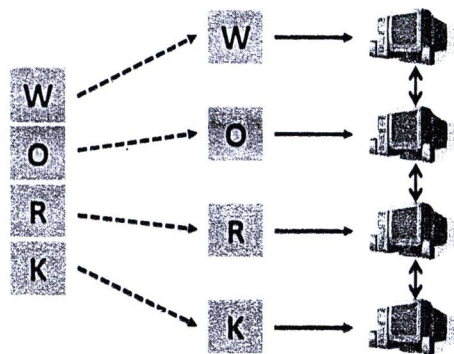
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

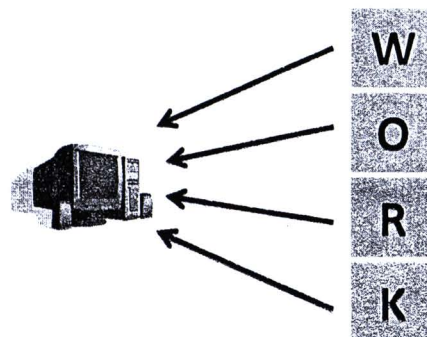
2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 การพัฒนาโปรแกรมคำนวณแบบขนาน (Parallel program development)

โปรแกรมคำนวณแบบขนาน (Parallel program) เป็นการใช้องค์ประกอบประมวลผลจำนวนมากช่วยกันทำงาน โดยมีเป้าหมายเพื่อให้โปรแกรมทำงานได้เร็วขึ้นหรือเพื่อให้สามารถทำงานที่มีขนาดใหญ่ขึ้น ด้วยการแบ่งปัญหาใหญ่ออกเป็นปัญหาย่อย ๆ หลาย ๆ ส่วน แล้วกระจายให้แต่ละองค์ประกอบประมวลผลหลาย ๆ เครื่องช่วยกันทำงานพร้อมกันดังรูปที่ 2-1 ซึ่งเป็นการพัฒนาการทำงานมาจากโปรแกรมคำนวณแบบลำดับ (Sequential program) ที่เป็นการทำงานด้วยการแก้ปัญหาทีละคำสั่งดังรูปที่ 2-2



รูปที่ 2-1 โปรแกรมคำนวณแบบขนาน (Parallel program)

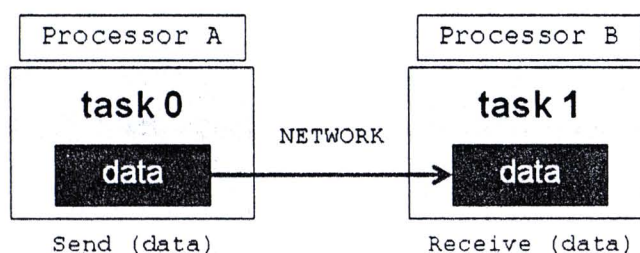


รูปที่ 2-2 โปรแกรมคำนวณแบบลำดับ (Sequential program)

สำหรับการพัฒนาโปรแกรมคำนวณแบบขนาน [1] ในงานวิจัยนี้มีรูปแบบการคำนวณแบบ Data parallel model เป็นหลัก และมีการเขียนโปรแกรมแบบที่ใช้การสื่อสารแบบ Message passing model ซึ่งมีรายละเอียดดังต่อไปนี้

2.1.1.1 Message Passing Model

เป็นแบบวิธีการสร้างโปรแกรมคำนวณแบบขนานที่ใช้โดยทั่วไป สำหรับการพัฒนาโปรแกรมคำนวณแบบขนานที่ทำงานบนระบบคลัสเตอร์คอมพิวเตอร์ ลักษณะของ Message passing model เป็นการใช้องค์ประกอบหลาย ๆ เครื่องที่มีหน่วยความจำเป็นของตัวเอง ช่วยกันทำงานไปพร้อมกันสำหรับปัญหาเดียวกัน โดยแต่ละหน่วยประมวลผลใช้การส่งข้อความระหว่างกันเพื่อแลกเปลี่ยนข้อมูลและประสานการทำงานขณะประมวลผลดังรูปที่ 2-3



รูปที่ 2-3 ลักษณะของ Message passing model

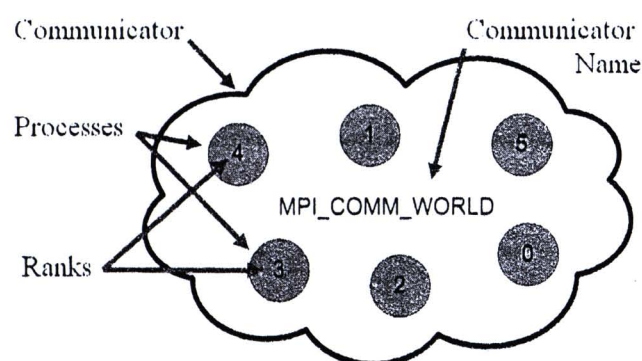
สำหรับการพัฒนาโปรแกรมแบบขนานด้วย Message passing model จะอาศัยไลบรารีของ Message Passing Interface (MPI) ที่สนับสนุนภาษาในการเขียนโปรแกรมได้หลายภาษา ซึ่งในงานวิจัยนี้ใช้การเขียนภาษา C++ ในการพัฒนาโปรแกรมคำนวณแบบขนาน

2.1.1.2 Message Passing Interface Library

Message Passing Interface หรือ MPI Library [2], [4], [5] เป็นชุดคำสั่งมาตรฐานของ Message Passing Model ซึ่งมีไว้สำหรับการสร้างโปรแกรมคำนวณแบบขนานด้วย Message Passing Model คือมีการติดต่อกันระหว่างหน่วยประมวลผลด้วยการส่งข้อความที่มีชุดคำสั่งที่เกี่ยวข้องกับการสื่อสารระหว่างเครือข่ายของแต่ละหน่วยประมวลผล ที่จะถูกระบุไว้อย่างชัดเจน (Explicit Parallelism) โดยโปรแกรมที่เขียนขึ้นด้วย MPI นั้น สามารถนำไปใช้งานบนระบบหรือเครื่องคอมพิวเตอร์ที่แตกต่างกันได้ง่าย

การทำงานของ MPI จะเป็นการแยกโปรแกรมออกเป็นกลุ่มงานให้ทำงานตามหน่วยประมวลผลย่อย ๆ ที่จะมีหมายเลขกำกับเฉพาะสำหรับแต่ละหน่วยประมวลผลหรือแต่ละกลุ่มงาน โดยจะไม่มีกรซ้ำกันของหมายเลขกำกับ ซึ่งใน MPI จะเรียกหมายเลขกำกับนี้ว่า Ranks

หรือ Process ID ที่จะมีค่าตั้งแต่ 0 ถึงจำนวนหน่วยประมวลผล ไปด้วยหนึ่ง (n-1) และมี Communicator เป็นตัวระบุหมายเลขกำกับให้กลุ่มงาน และดำเนินการติดต่อหรือส่งข้อความถึงกันระหว่างหน่วยประมวลผล สำหรับโปรแกรมพื้นฐานทั่วไปจะใช้ MPI_COMM_WORLD เป็นชื่อ Communicator สำหรับการติดต่อหรือส่งข้อความกันระหว่างหน่วยประมวลผลสำหรับทุกกลุ่มงานของโปรแกรกดังรูปที่ 2-4



รูปที่ 2-4 จำลองการควบคุมกลุ่มงานหรือหน่วยประมวลผลของ MPI_COMM_WORLD

การใช้งาน MPI สามารถใช้งานกับชุดคำสั่งภาษา C++ และชุดคำสั่งภาษา FORTRAN ซึ่งในโปรแกรมจำลองสื่อนำด้วยการคำนวณแบบขนานนี้จะใช้งาน MPI ผ่านชุดคำสั่งภาษา C++ โดยนำเข้าแฟ้มข้อมูล mpi.h ในส่วนหัวของโปรแกรม เพื่อเรียกใช้งานฟังก์ชันต่าง ๆ ของ MPI โดยฟังก์ชันที่นำมาใช้งานในโปรแกรม มีดังต่อไปนี้

1. MPI_Init – ใช้เพื่อเริ่มต้นการทำงานของโปรแกรมที่ใช้งาน MPI เพื่อเริ่มการทำงานของ Library และกำหนดค่า Environment ต่าง ๆ ของ MPI
2. MPI_Finalize – ใช้เพื่อสิ้นสุดการทำงานของโปรแกรมที่ใช้งาน MPI โดยส่งสัญญาณให้ MPI runtime library จบการทำงานของ MPI อย่างสมบูรณ์แบบ
3. MPI_Abort – ใช้เพื่อยกเลิกการทำงานของโปรแกรมที่ใช้งาน MPI ให้หยุดการทำงานทุก ๆ หน่วยประมวลผล ซึ่งจะเรียกใช้งานใช้ ก็ต่อเมื่อโปรแกรมไม่สามารถเริ่มต้นการทำงานของ MPI ได้อย่างสมบูรณ์
4. MPI_Comm_size – ใช้เพื่อรับค่าจำนวนของหน่วยประมวลผลทั้งหมดที่ใช้งานด้วย MPI

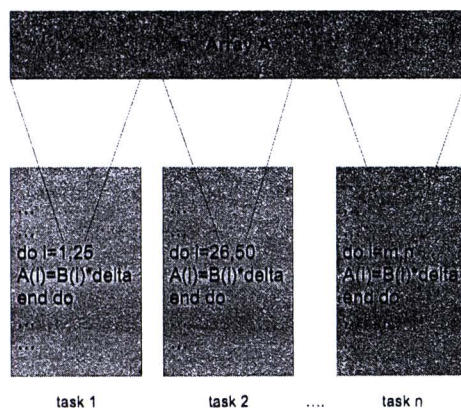
5. MPI_Comm_rank – ใช้เพื่อรับค่าหมายเลขกำกับของหน่วยประมวลผล เพื่อระบุส่วนของงานที่จะให้แต่ละหน่วยประมวลผลทำงาน
6. MPI_Barrier – ใช้เพื่อให้การทำงานของหน่วยประมวลผลหยุดการทำงาน และจะเริ่มทำงานอีกต่อไปพร้อมกันในทุกหน่วยประมวลผล เมื่อมีการเรียกใช้งานฟังก์ชัน MPI_Barrier นี้ครบทุกหน่วยประมวลผล
7. MPI_Wtime – ใช้เพื่อรับค่าเวลาในหน่วยวินาทีของหน่วยประมวลผลที่ถูกใช้งานโดย MPI
8. MPI_Isend – ใช้ในการส่งข้อความถึงบัฟเฟอร์ของหน่วยประมวลผลปลายทางแบบ non-blocking คือเมื่อหน่วยประมวลผลทำการส่งข้อความไปยังปลายทางแล้ว ก็จะทำงานต่อไปได้ทันที โดยไม่รอว่าการส่งข้อความนี้จะเสร็จเรียบร้อยหรือไม่
9. MPI_Irecv – ใช้ในการรับข้อความจากบัฟเฟอร์ของหน่วยประมวลผลต้นทางแบบ non-blocking คือเมื่อหน่วยประมวลผลได้รับข้อความจากต้นทางแล้ว ก็จะทำงานต่อไปได้ทันที โดยไม่รอว่าการส่งข้อความนี้จะเสร็จเรียบร้อยหรือไม่
10. MPI_Wait – ใช้สำหรับรอให้ MPI ที่ส่งหรือรับข้อความแบบ non-blocking ทำงานจนเสร็จสมบูรณ์
11. MPI_Waitall – ใช้สำหรับรอให้ MPI ทำงานในทุก ๆ การเชื่อมต่อเสร็จสมบูรณ์

2.1.1.3 Data parallel model

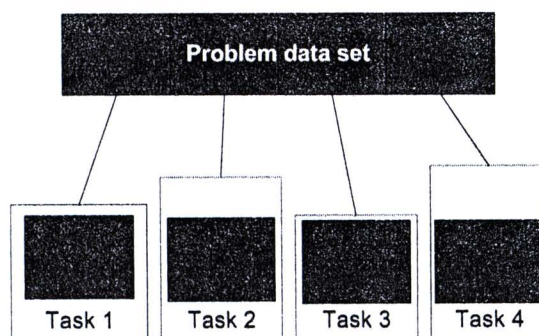
เป็นแบบวิธีการสร้างโปรแกรมคำนวณแบบขนานที่เน้นปัญหาที่มีการคำนวณแบบเดียวกันบนชุดข้อมูลที่มีอยู่ในรูปแบบอาร์เรย์ (Array) โดยใช้การแบ่งย่อยชุดข้อมูลออกเป็น ส่วน ๆ เพื่อส่งให้แต่ละหน่วยประมวลผลทำการคำนวณดังรูปที่ 2-5 ซึ่งในงานวิจัยนี้ชนิดของปัญหาที่มีความเป็นอิสระของข้อมูล (Data independence) จึงสามารถนำวิธีการนี้มาใช้ในการแบ่งปัญหาออกเป็น ส่วน ๆ ซึ่งประกอบด้วย 2 วิธีการ ดังนี้

1. การแบ่งด้วยขอบเขตของปัญหา (Domain Decomposition) เป็นการแบ่งข้อมูลของปัญหาออกเป็น ส่วน ๆ ให้มีขนาดเล็กลงตามลักษณะของข้อมูล เพื่อแบ่งให้แต่ละหน่วยประมวลผลช่วยกันทำงานดังรูปที่ 2-6

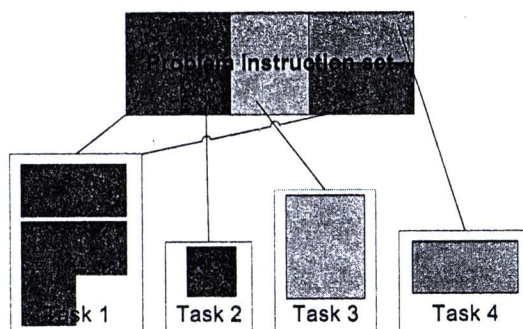
2. การแบ่งด้วยวิธีการทำงาน (Functional Decomposition) เป็นการแบ่งการคำนวณที่ต่างกันของปัญหาออกเป็นส่วน ๆ เพื่อแบ่งให้แต่ละหน่วยประมวลผลช่วยกันทำงานดังรูปที่ 2-7



รูปที่ 2-5 ลักษณะของ Data parallel model



รูปที่ 2-6 ลักษณะการแบ่งด้วยของเซตของปัญหา



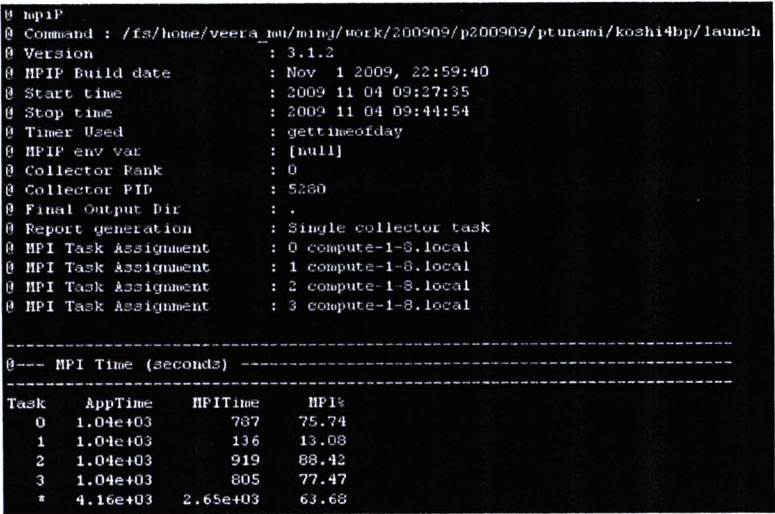
รูปที่ 2-7 ลักษณะการแบ่งด้วยวิธีการทำงาน

2.1.2 การเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนาน (MPI Profiling)

การเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนานนี้ เป็นการเก็บข้อมูลเวลาที่ใช้ในการทำงานของโปรแกรมในเชิงสถิติ เพื่อนำมาวิเคราะห์ประสิทธิภาพการทำงานของแต่ละส่วนการทำงานในโปรแกรมว่าส่วนการทำงานแต่ละส่วนใช้เวลาในการทำงานมากน้อยเพียงใดในแต่ละรอบการคำนวณ ซึ่งการเก็บข้อมูลการทำงานนี้เป็นกระบวนการที่จะช่วยให้ทราบปัญหาของโปรแกรมที่สามารถปรับแต่งประสิทธิภาพได้ โดยแนวทางในการเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนาน ในงานวิจัยนี้จะใช้งานชุดการเก็บข้อมูลการทำงานของโปรแกรม 2 โปรแกรมดังนี้

2.1.2.1 MpiP Version 3.1.2 (Lightweight, Scalable MPI Profiling) and MpiPView Version 2.02

MpiP [6], [7] เป็นชุดการทำงานขนาดเล็กสำหรับโปรแกรมคำนวณแบบขนานด้วย MPI ทำการเก็บข้อมูลเชิงสถิติเกี่ยวกับการทำงานของฟังก์ชัน MPI ในโปรแกรมที่เขียนด้วยชุดคำสั่งภาษา C++ สามารถนำมาติดตั้งและใช้งานได้สะดวกในทรัพยากรที่แตกต่างกันอย่างระบบคลัสเตอร์ TERA และระบบคลัสเตอร์ TSUBAME โดยผลที่ได้จาก MpiP นี้ถูกรายงานออกมาเป็นแฟ้มข้อมูลที่มีรายละเอียดของการทำงานในแต่ละส่วนของการเรียกใช้งานฟังก์ชัน MPI ต่าง ๆ ของโปรแกรม ซึ่งการเก็บข้อมูลการทำงานของโปรแกรมจำลองสึนามิแบบขนานนี้ ฟังก์ชัน MPI ที่ถูกเก็บผลการทำงานขึ้นมามีดังนี้คือ MPI_Barrier MPI_Isend MPI_Irecv MPI_Wait และ MPI_Waitall ซึ่งถูกบันทึกและรายงานผลการทำงานของโปรแกรมออกมาเป็นรูปแบบของข้อความที่สามารถนำมาวิเคราะห์ผลได้ดังรูปที่ 2-8



รูปที่ 2-8 ตัวอย่างการบันทึกผลจากชุดการเก็บข้อมูลการทำงานของโปรแกรมด้วย MpiP



ข้อมูลที่วิเคราะห์ได้จากผลการเก็บข้อมูลด้วย MpiP คือ ระยะเวลาที่ใช้ในการทำงานระหว่างการทำงานของฟังก์ชัน MPI กับการทำงานของโปรแกรม จำนวนการเรียกใช้งานฟังก์ชัน MPI ระยะเวลาที่ใช้ในแต่ละครั้งของการเรียกใช้งานฟังก์ชัน MPI และปริมาณการสื่อสารกันของแต่ละครั้งที่เรียกใช้งานฟังก์ชัน MPI โดยรายงานผลของการทำงานแยกกันในแต่ละหน่วยประมวลผล แต่ในการศึกษาและวิเคราะห์ผลที่จะได้รับความสะดวกมากขึ้นนั้น สามารถใช้โปรแกรม MpiPView [8] มาช่วยในการดูผลของการเก็บข้อมูลการทำงานจาก MpiP เพื่อให้การวิเคราะห์ผลทำได้ง่ายขึ้น โดยโปรแกรมทำการจัดการแบ่งผลของการเก็บข้อมูลเป็นส่วน ๆ ที่สามารถเลือกดูได้ และเรียงลำดับของผลให้ตามสัดส่วนของระยะเวลาที่ฟังก์ชัน MPI ทำงานจากมากไปน้อย ที่สามารถดูรายละเอียดย่อยของการทำงานนั้น ๆ ในแต่ละงานที่แบ่ง พร้อมทั้งแสดงถึงตำแหน่งของการเรียกใช้งานฟังก์ชัน MPI จากในโค้ดของโปรแกรม ตามรูปที่ 2-9

Task	Count	Max (ms)	Mean (ms)	Min (ms)	MP %	App %
ALL	1396000	1120.0000	14.4000	0.0160	8.91	9.78
3	72000	1080.0000	5.6000	0.0160	4.05	3.28
6	78000	1040.0000	3.6000	0.0170	8.99	8.12
9	70000	1100.0000	30.0000	0.0160	19.97	12.11
7	78000	1020.0000	2.7000	0.0170	6.66	6.19
8	72000	1030.0000	1.1100	0.0190	1.05	0.64

```
3510  // Send T TIMING
3511  // time = MPI_Wtime();
3520  // Send T TIMING
3521  // Receive m1, m2 from m3
3522  {
3523      for (int i = 0; i < m1-numbNodesInRing; i++)
3524          AllwRecv(1, MPI_INT, MPI_ANY_SOURCE, &recvBuf[i], &recvStatus[i]);
3525  }
3526  for (int i = 0; i < m2-numbNodesInRing; i++)
3527      AllwRecv(1, MPI_INT, MPI_ANY_SOURCE, &recvBuf[i], &recvStatus[i]);
3528  }
3529  // End of T TIMING
3530  // End of T TIMING
```

รูปที่ 2-9 ตัวอย่างการรายงานผลจากการเก็บข้อมูลด้วยโปรแกรม MpiPView

2.1.2.2 IPM Version 0.980 (Integrated Performance Monitoring) and Ploticus Version 2.41

IPM [9], [10] เป็นชุดการเก็บข้อมูลการทำงานของโปรแกรมคำนวณแบบขนานที่สามารถใช้งานได้ง่าย และมีขนาดเล็ก สามารถติดตั้งลงบนระบบคลัสเตอร์ TERA และระบบคลัสเตอร์ TSUBAME เพื่อใช้ทำการเก็บผลการการทำงานของโปรแกรมคำนวณแบบขนานที่ใช้ MPI ในการควบคุมการทำงาน โดยผลการการทำงานของโปรแกรมที่ได้จาก IPM จะแตกต่างกับ MpiP ตรงที่ผลการรายงานที่ได้จาก IPM จะออกมาเป็นรูปแบบของแฟ้มข้อมูล XML ซึ่งจะเป็นดังรูปที่ 2-10 และจะใช้โปรแกรม Ploticus [11] ในการแปลงข้อมูลของผลการการทำงานของโปรแกรมที่ได้ ให้

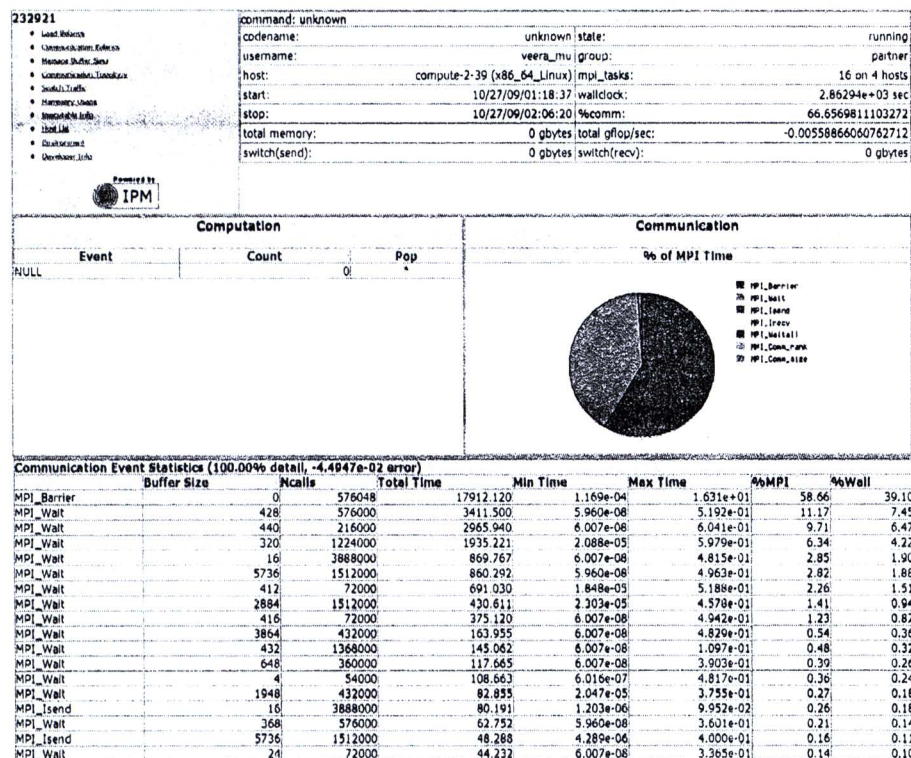
สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุด
วันที่..... 7 ส.ค. 2555
เลขทะเบียน..... 247994
เลขเรียกหนังสือ.....



ออกมาเป็นรูปภาพที่แสดงเป็นแฟ้มข้อมูลเว็บเพจ เพื่อให้ทำการวิเคราะห์ข้อมูลได้สะดวกขึ้น โดย
การแสดงผลจะเป็นไปตามตัวอย่างดังรูปที่ 2-11 และรูปที่ 2-12

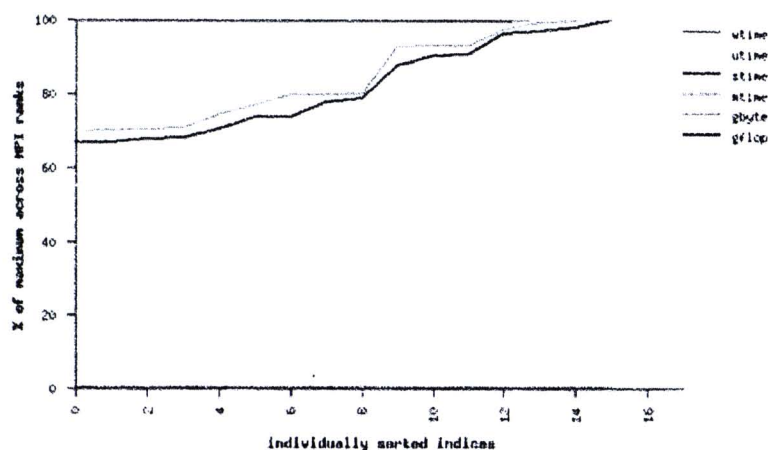
```
<?xml version="1.0" encoding="iso-8859-1" ?>
<ipm_job_profile>
<task ipm_version="0.980" cookie="1256581117.688157" mpi_rank="0" mpi_size="16" stamp_init
="1256581117.688157" stamp_final="1256583980.609330" username="veera_m" groupname="partne
r" flags="688063600" pid="6609" >
<job nhosts="4" ntasks="16" start="1256581117" final="1256583980" cookie="1256581117.68815
7" code="unknown" >232921</job>
<host mach_name="unknown" mach_info="x86_64_linux" >compute-2-39</host>
<perf wtime="2.86292e+03" utime="7.04320e+03" stime="1.824e+03" mtime="2.25221e+03" gflop=
"-1.00000e+00" gbyte="0.00000e+00" ></perf>
<switch bytes_tx="0.00000e+00" bytes_rx="0.00000e+00" > </switch>
<region label="ipm_noregion" nexecs="3" wtime="4.8503e+06" utime="6.6270e+02" stime="1.757
5e+03" mtime="2.2603e+03" >
  <hpm api="PIS&BL&ED" ncounter="1" eventset="0" gflop="-1.0000e-09" >
<counter name="NULL" > 0 </counter>
</hpm>
<func name="MPI_Comm_rank" count="1" > 1.0245e-07 </func>
<func name="MPI_Comm_size" count="1" > 1.0303e-07 </func>
<func name="MPI_Isend" count="45000" > 1.5162e+00 </func>
<func name="MPI_Irecv" count="45000" > 2.6424e-01 </func>
<func name="MPI_Wait" count="45000" > 6.0878e+01 </func>
<func name="MPI_Waitall" count="45000" > 4.0152e-02 </func>
<func name="MPI_Barrier" count="36003" > 2.1977e+03 </func>
</region>
```

รูปที่ 2-10 ตัวอย่างการบันทึกผลจากชุดการเก็บข้อมูลการทำงานของโปรแกรมด้วย IPM



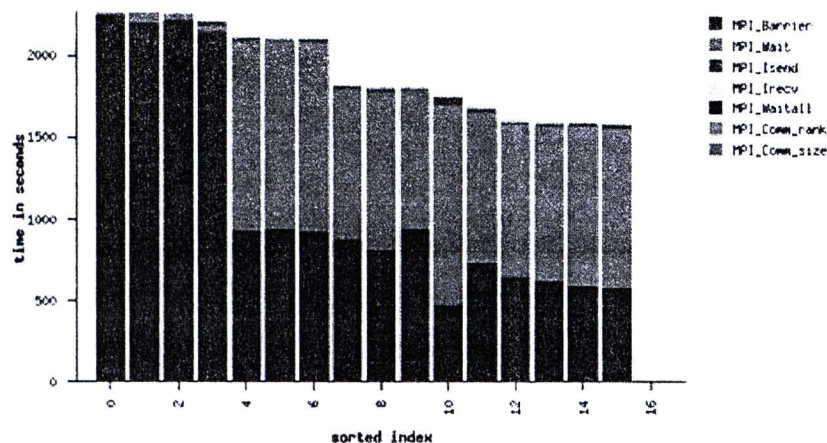
รูปที่ 2-11 ตัวอย่างของการรายงานผลการดำเนินงานของโปรแกรมคำนวณแบบขนานด้วย IPM

Load balance by task memory, flops, timings



by MPI rank, by MPI time

Communication balance by task (sorted by MPI time)



by MPI rank, time detail by MPI time, time detail by rank, call list

รูปที่ 2-12 ตัวอย่างของการรายงานผลการทำงานของโปรแกรมคำนวณแบบขนานด้วย IPM (ต่อ)

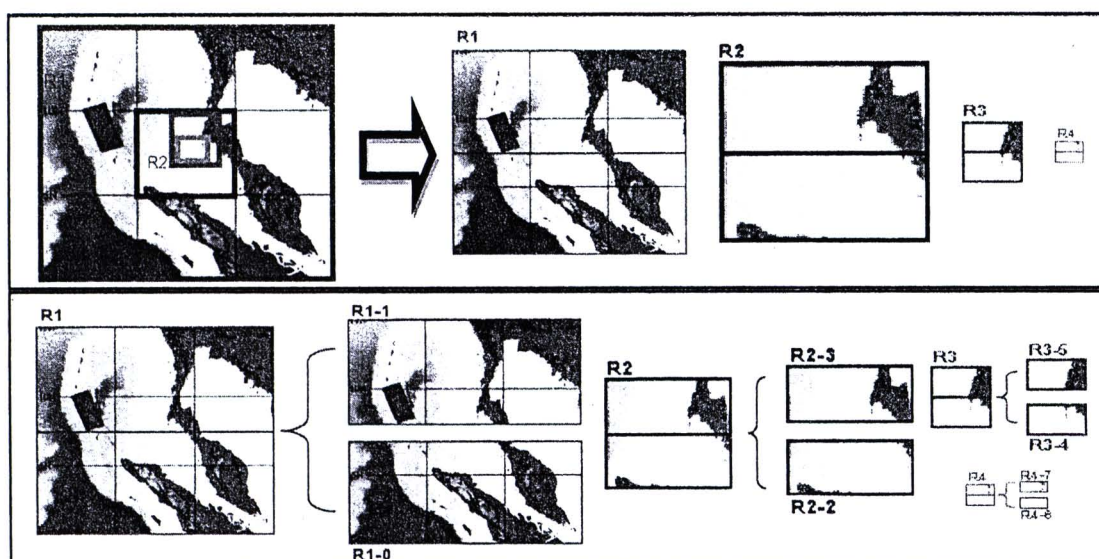
นอกจากการใช้งาน MpiP และ IPM ในการเก็บข้อมูลการทำงานของโปรแกรมแล้ว จะทำการแทรกฟังก์ชัน MPI_Wtime ลงในโปรแกรม เพื่อให้ได้เวลาต้นและปลายของการทำงานในส่วนที่ต้องการเก็บระยะเวลาในการทำงาน ผลต่างของเวลาก็จะได้เป็นระยะเวลาที่ใช้ในการทำงานของแต่ละส่วน ซึ่งผลที่ได้จากการแทรกฟังก์ชันดังกล่าวจะทำให้ทราบถึงระยะเวลาการทำงานของแต่ละส่วนในโปรแกรม และนำไปวิเคราะห์เพื่อทำการปรับปรุงแบบของการจัดการงานและปรับปรุงประสิทธิภาพต่อไป

2.1.3 การปรับแต่งและวัดประสิทธิภาพของโปรแกรมคำนวณแบบขนาน (Performance Tuning)

การปรับแต่งประสิทธิภาพของโปรแกรม สามารถจัดการได้ในส่วนของรูปแบบการแบ่งงานที่ถูกส่งไปคำนวณแบบขนานจากในแฟ้มจัดเก็บข้อมูล โดยเน้นที่วิธีการแบ่งงานและการกระจายงานให้หน่วยประมวลผลอย่างสม่ำเสมอ เพื่อลดปริมาณการสื่อสารและการประสานการทำงานระหว่างหน่วยประมวลผล ซึ่งจะใช้การปรับตัวแปรต่าง ๆ เช่น จำนวนหน่วยประมวลผล ลักษณะการแบ่งงานและการกระจายงาน การประสานการทำงานและการสื่อสารระหว่างหน่วยประมวลผล ให้เหมาะสมกับขนาดของงาน ลักษณะการทำงานของโปรแกรม และขีดความสามารถของระบบคลัสเตอร์คอมพิวเตอร์ที่นำมาใช้ในการคำนวณแบบขนาน ส่วนการวัดประสิทธิภาพของโปรแกรมคำนวณแบบขนานนี้วัดจากอัตราเร็วที่เพิ่มขึ้น (Speedup) และค่าประสิทธิภาพของการทำงาน (Efficiency)

2.1.3.1 วิธีการแบ่งงานและการกระจายงาน (Partitioning and Load Balancing)

ในการคำนวณแบบขนานนั้น จะจำเป็นต้องแบ่งข้อมูลที่จะทำการคำนวณออกเป็น ส่วน ๆ เพื่อให้แต่ละหน่วยประมวลผลทำการคำนวณแต่ละส่วนไปพร้อม ๆ กัน โดยวิธีการแบ่งงานนั้นสามารถทำได้ทั้งตามคอลัมน์และตามแถว แต่ในงานวิจัยนี้จะใช้การแบ่งข้อมูลตามแถวมาใช้ในการหารูปแบบของการจัดการงาน เนื่องจากทำการคำนวณด้วยโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานได้รวดเร็วกว่าการแบ่งข้อมูลตามคอลัมน์ อีกทั้งสามารถนำข้อมูลมาเชื่อมต่อกันได้ง่ายและรวดเร็วกว่าการแบ่งข้อมูลตามคอลัมน์



รูปที่ 2-13 ตัวอย่างการแบ่งข้อมูลตามแถวเพื่อใช้จำลองสึนามิด้วยการคำนวณแบบขนาน

โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานนี้ สามารถปรับแต่งการแบ่งงานในแต่ละระดับความละเอียดได้ โดยจากรูปที่ 2-13 พื้นที่ในทุกระดับความละเอียดได้ถูกแบ่งการทำงานออกเป็นสองส่วน ที่แต่ละส่วนนั้นจะถูกคำนวณด้วยหน่วยประมวลผลแยกกันตามส่วน ซึ่งหากข้อมูลมีจำนวนพื้นที่ทั้งหมด 4 ระดับ และแต่ละส่วนไม่มีการแบ่งพื้นที่คำนวณเลย ก็ใช้หน่วยประมวลผลทั้งหมด 4 หน่วย แต่ถ้าในแต่ละระดับความละเอียดแบ่งออกเป็นสองส่วน ก็ใช้หน่วยประมวลผลทั้งหมด 8 หน่วย โดยการทำงานของโปรแกรมนี้สามารถระบุได้ว่าต้องการแบ่งข้อมูลในแต่ละระดับความละเอียดออกเป็นกี่ส่วน และสามารถเลือกที่จะเพิ่มจำนวนพื้นที่ที่สนใจในแต่ละระดับความละเอียดได้ ในการคำนวณเพื่อจำลองสึนามิแต่ละครั้ง จากนั้นแฟ้มจัดเก็บข้อมูล

โดยรูปแบบของการแบ่งงานที่จะทำให้เกิดการกระจายงานอย่างสม่ำเสมอของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานนี้ จำเป็นต้องคำนึงถึงปัจจัยที่มีความซับซ้อนของข้อมูลที่นำมาคำนวณ และส่งผลต่อความเร็วในการทำงานของโปรแกรม ซึ่งมีดังนี้คือ

1. ขนาดของพื้นที่ในแต่ละระดับความละเอียดที่มีขนาดไม่เท่ากัน ทำให้การแบ่งงานในแต่ละระดับความละเอียดต้องปรับเปลี่ยนตามขนาดของพื้นที่ที่สนใจ
2. ความละเอียดของแต่ละระดับความละเอียดที่มีความแตกต่างกัน โดยที่พื้นที่ระดับ R4 มีความละเอียดเป็น 3 เท่าของพื้นที่ระดับ R3 ที่พื้นที่ระดับ R3 มีความละเอียดเป็น 3 เท่าของพื้นที่ระดับ R2 และที่พื้นที่ระดับ R2 มีความละเอียดเป็น 8 เท่าของพื้นที่ระดับ R1
3. หลักการที่ใช้ในการคำนวณที่แตกต่างกันระหว่างระดับความละเอียดของพื้นที่ระดับ R1 ที่ใช้การคำนวณด้วยสมการเชิงเส้นด้วยการจำลองระยะไกล กับระดับความละเอียดของพื้นที่ระดับ R2 R3 และ R4 ที่ใช้การคำนวณด้วยสมการไม่เชิงเส้นด้วยการจำลองระยะใกล้
4. จำนวนรอบของการคำนวณที่มีความแตกต่างกัน โดยการคำนวณของการจำลองระยะไกล ในส่วนการคำนวณหลักมีการเรียกใช้งานในทุก ๆ 4 รอบการคำนวณ ยกเว้นส่วนการคำนวณผลให้การคำนวณของการจำลองระยะใกล้ที่มีการเรียกใช้งานทุก ๆ รอบการคำนวณ ขณะที่ในการคำนวณของการจำลองระยะไกล ในทุกส่วนการคำนวณมีการเรียกใช้งานทุก ๆ รอบการคำนวณ



ฉะนั้นการแบ่งงานและกระจายงานให้สม่ำเสมอ นั้น จำเป็นต้องหาความสัมพันธ์ของปัจจัยต่าง ๆ ของโปรแกรมจำลองสินามิตด้วยการคำนวณแบบขนานให้ได้ ซึ่งใช้การแบ่งงานที่ดูจากขนาดของพื้นที่ในการคำนวณ ที่แบ่งให้แต่ละหน่วยประมวลผลทำงานเท่า ๆ กัน โดยต้องหาอัตราการแบ่งงานระหว่างการจำลองระยะไกล และการจำลองระยะใกล้ ออกมาเป็นระเบียบวิธีการจัดการงานสำหรับงานที่มีความซับซ้อนของพื้นที่ในการคำนวณ

2.1.3.2 อัตราเร็วที่เพิ่มขึ้น (Speedup)

ในการทดสอบประสิทธิภาพของการคำนวณแบบขนานที่เพิ่มขึ้นเมื่อเทียบกับการคำนวณแบบลำดับ สามารถทำได้โดยการใช้อัตราความเร็วที่เพิ่มขึ้น (S_p) [12] ที่หาได้จาก

$$S_p = \frac{T_s}{T_p} \quad \text{เมื่อ}$$

- S_p อัตราเร็วที่เพิ่มขึ้นเมื่อใช้ p หน่วยประมวลผล โดยค่า S_p ยิ่งมากยิ่งดี
 p จำนวนหน่วยประมวลผลที่ใช้ในการคำนวณ
 T_s เวลาที่ใช้ในการทำงานของการคำนวณแบบลำดับ
 T_p เวลาที่ใช้ในการทำงานของการคำนวณแบบขนานเมื่อใช้ p หน่วยประมวลผล

2.1.3.3 ประสิทธิภาพของการทำงาน (Efficiency)

จากค่าอัตราเร็วที่เพิ่มขึ้น จะสามารถหาค่าประสิทธิภาพของการทำงานได้จาก

$$E_p = \frac{S_p}{p} = \frac{T_s}{p T_p} \quad \text{เมื่อ}$$

- E_p ประสิทธิภาพของการทำงาน โดยค่า E_p มียิ่งมากยิ่งดี
 S_p อัตราเร็วที่เพิ่มขึ้นเมื่อใช้ p หน่วยประมวลผล
 p จำนวนหน่วยประมวลผลที่ใช้ในการคำนวณ
 T_s เวลาที่ใช้ในการทำงานของการคำนวณแบบลำดับ
 T_p เวลาที่ใช้ในการทำงานของการคำนวณแบบขนานเมื่อใช้ p หน่วยประมวลผล

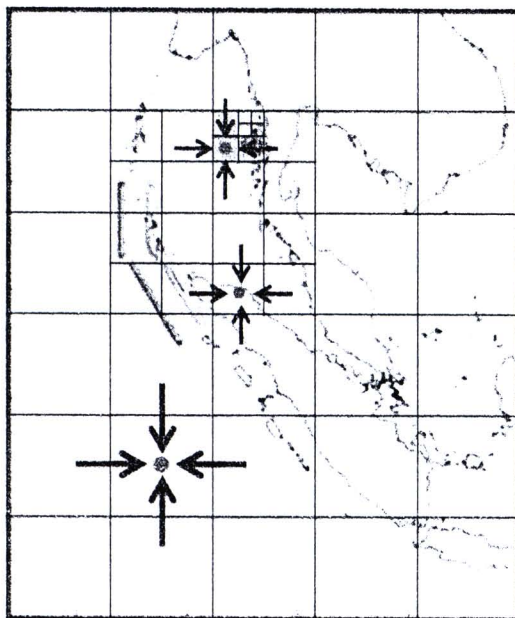
2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 โปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ (TUNAMI program)

โปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ หรือโปรแกรมทูนามิ (Tohoku University's Numerical Analysis Model for Investigation of tsunamis หรือ TUNAMI) [13] เป็นโปรแกรมจำลองสึนามิที่เขียนด้วยชุดคำสั่งภาษา FORTRAN (FORmula TRANslation) และเป็นโปรแกรมที่ใช้การคำนวณแบบลำดับ (Sequential Program) ซึ่งเป็นผลงานวิจัยของ Dr.Fumihiko Imamura แห่งมหาวิทยาลัยโตโฮกุ (Tohoku University) ประเทศญี่ปุ่น กับ Dr. Ahmet Cevdet Yalciner และผู้ช่วยนักวิจัย Gulizar Ozyurt แห่งมหาวิทยาลัยมิดเดิล อีสต์ เทคนิคัล ประเทศตุรกี และผศ.ดร.อาณัติ เรืองรัมย์ ศูนย์เชี่ยวชาญเฉพาะทางด้านวิศวกรรมแผ่นดินไหว และการสั่นสะเทือน ภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย เป็นผู้นำมาปรับปรุงและพัฒนาต่อเพื่อนำมาใช้ในการเตือนภัยสึนามิจากเหตุการณ์แผ่นดินไหวในเขตมหาสมุทรอินเดีย โดยเมื่อนำมาใช้งานแล้วถือว่าเป็นโปรแกรมจำลองสึนามิ ที่ได้ผลการคำนวณที่ถูกต้องและแม่นยำ เป็นที่น่าพอใจที่สุด แต่โปรแกรมนี้อาจใช้เวลาในการจำลองสึนามิแต่ละกรณีประมาณ 4 - 6 ชั่วโมง ทั้งนี้จะขึ้นกับขนาดของแต่ละระดับความละเอียดของพื้นที่ที่ใช้ในการคำนวณ และสมรรถนะของเครื่องคอมพิวเตอร์ที่ใช้ โดยการนำโปรแกรมจำลองสึนามิไปใช้มีด้วยกัน 2 วิธี ซึ่งวิธีแรกเป็นการนำไปคำนวณกรณีที่เกิดขึ้นได้จำนวนมาก แล้วเก็บผลลัพธ์ไว้เป็นฐานข้อมูลเพื่อนำมาประกอบการตัดสินใจเตือนภัย เมื่อเกิดแผ่นดินไหวในกรณีที่ใกล้เคียงกับที่เคยจำลองไว้ ซึ่งกรณีต่าง ๆ ที่ต้องทำการจำลองนั้นมีจำนวนมากกว่า 1,000 กรณี ส่วนวิธีที่สองเป็นการนำไปคำนวณทันทีที่เกิดแผ่นดินไหว เพื่อประกอบการตัดสินใจเตือนภัยที่แม่นยำกว่าที่ได้จากกรณีที่คำนวณไว้ล่วงหน้า ซึ่งวิธีนี้จำเป็นต้องคำนวณให้เร็ว

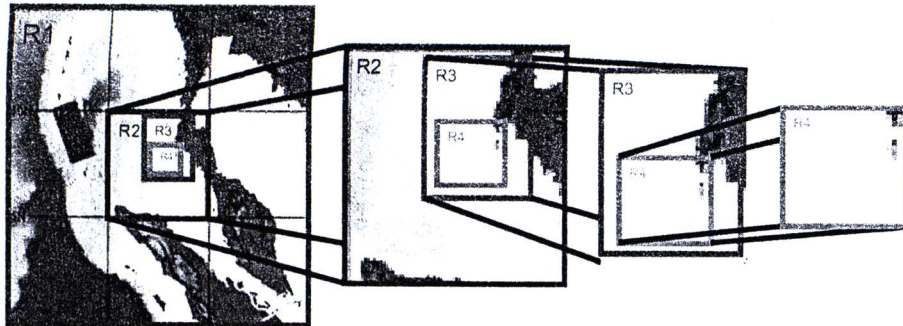
ในการคำนวณของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับนี้ ได้แบ่งการคำนวณออกเป็น 2 ส่วน คือ การจำลองคลื่นสึนามิระยะไกลในระบบพิกัดทรงกลม หรือการจำลองระยะไกล (Far-field Tsunami Simulation in Spherical Coordinate System) และการจำลองคลื่นสึนามิระยะใกล้ในระบบพิกัดคาร์ทีเซียน หรือการจำลองระยะใกล้ (Near-field Tsunami Simulation in Cartesian Coordinate System) ด้วยการคำนวณที่ใช้ระเบียบวิธีการวิเคราะห์เชิงตัวเลขชื่อว่า Finite Difference Method ที่การหาคำตอบของตำแหน่งที่สนใจนั้น หาได้จากคำตอบของบริเวณใกล้เคียงตำแหน่งที่สนใจ ณ เวลาก่อนหน้า ดังที่แสดงให้เห็นจากรูปที่ 2-14 เนื่องจากในแต่ละบริเวณมีความต้องการความละเอียดของข้อมูลที่แตกต่างกัน เช่น พื้นที่ชายฝั่งต้องการความละเอียดในการคำนวณมากกว่าพื้นที่กลางมหาสมุทร และเมื่อความละเอียดของ

ข้อมูลสูงขึ้น ก็ใช้เวลาในการคำนวณก็มากขึ้นด้วย ดังนั้นโปรแกรม TUNAMI จึงใช้วิธีการคำนวณแบบหลายระดับความละเอียด (Multi-scale) ที่มีการแลกเปลี่ยนข้อมูลกันระหว่างการคำนวณในแต่ละระดับความละเอียดที่บริเวณรอยต่อของพื้นที่ เพื่อให้ได้การคำนวณที่ถูกต้องแม่นยำ และมีความต่อเนื่องกันของข้อมูลในแต่ละระดับความละเอียด



รูปที่ 2-14 การหาคำตอบของตำแหน่งใด ๆ ของโปรแกรม

โดยในแต่ละระดับความละเอียดจะถูกเรียกว่า Region ซึ่งจะใช้สัญลักษณ์ R1, R2, R3 และ R4 แทนแต่ละระดับความละเอียดดังรูปที่ 2-15 เป็นการแสดงตัวอย่างของข้อมูลบริเวณที่นำมาคำนวณ ที่มีด้วยกันทั้งหมด 4 ระดับความละเอียด ซึ่งเมื่อทำการคำนวณเพื่อจำลองการเกิดคลื่นสึนามิแล้ว ผลการจำลองที่ได้ตามความละเอียดของการคำนวณที่แตกต่างกันนี้ขึ้นกับลักษณะการนำไปใช้งานในการเตือนภัย โดยพื้นที่ที่มีระดับความละเอียดมากที่สุดนั้น เมื่อระบุให้โปรแกรมทำการคำนวณเพื่อให้น้ำทะเลท่วมขึ้นชายฝั่งทะเลได้ (Run-up) ภาพการจำลองสึนามิที่ได้ออกมานั้น จะสามารถบอกถึงความเสียหายที่จะเกิดขึ้นกับพื้นที่ชายฝั่งได้อย่างแม่นยำ ทั้งนี้จะเรียกพื้นที่บริเวณที่มีความละเอียดมากที่สุด หรือพื้นที่ระดับ R4 ว่า "โซน" ซึ่งโปรแกรมนี้นี้เป็นการคำนวณที่เจาะจงไปบนพื้นที่เพียงจุดเดียวในพื้นที่ระดับ R4 ที่มีพื้นที่ระดับ R3 R2 และ R1 ครอบคลุมอยู่ เมื่อต้องการเปลี่ยนการคำนวณในพื้นที่ระดับ R4 ไปยังตำแหน่งอื่น โปรแกรมจำลองสึนามิก็ต้องทำการคำนวณใหม่ตามพื้นที่ที่เปลี่ยนไป



รูปที่ 2-15 ตัวอย่างระดับความละเอียดของข้อมูลบริเวณที่คำนวณด้วยโปรแกรมจำลองสึนามิ

เนื่องจากโปรแกรม TUNAMI สามารถทำงานได้เพียงที่ละโซนดังรูปที่ 2-15 ดังนั้นหากต้องการทำการจำลองสึนามิในหนึ่งกรณีของการเกิดแผ่นดินไหวใต้ทะเล โดยให้ครอบคลุมพื้นที่ดังที่กำหนดในตารางที่ 2-1 จำเป็นต้องทำการคำนวณพื้นที่ทั้งหมด 18 โซน ดังนี้

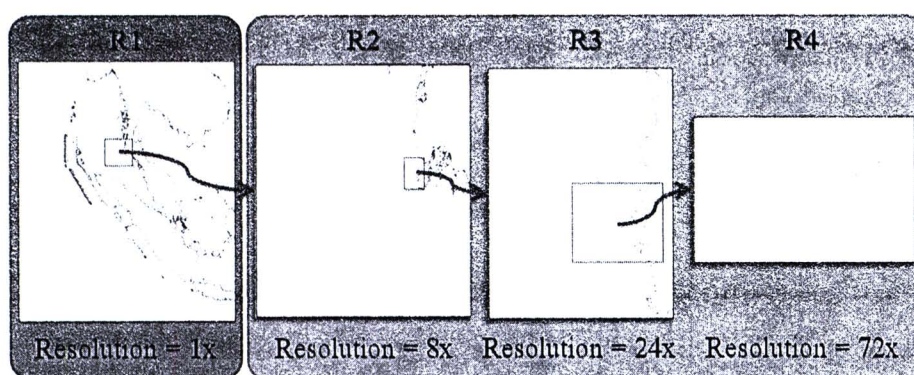
1. โซน 2111 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 21 > 211 > 2111
2. โซน 2112 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 21 > 211 > 2112
3. โซน 2121 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 21 > 212 > 2121
4. โซน 2122 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 21 > 212 > 2122
5. โซน 2131 ใช้ข้อมูลในระดับความละเอียดหมายเลข 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. โซน 2213 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 22 > 221 > 2213
13. โซน 2311 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 231 > 2311
14. โซน 2312 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 231 > 2312
15. โซน 2313 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 231 > 2313
16. โซน 2321 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 232 > 2321
17. โซน 2322 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 232 > 2322
18. โซน 2323 ใช้ข้อมูลในระดับความละเอียดหมายเลข 1 > 23 > 232 > 2323

ตารางที่ 2-1 พิกัดของบริเวณที่สนใจที่จะนำมาคำนวณด้วยโปรแกรมจำลองสึนามิ

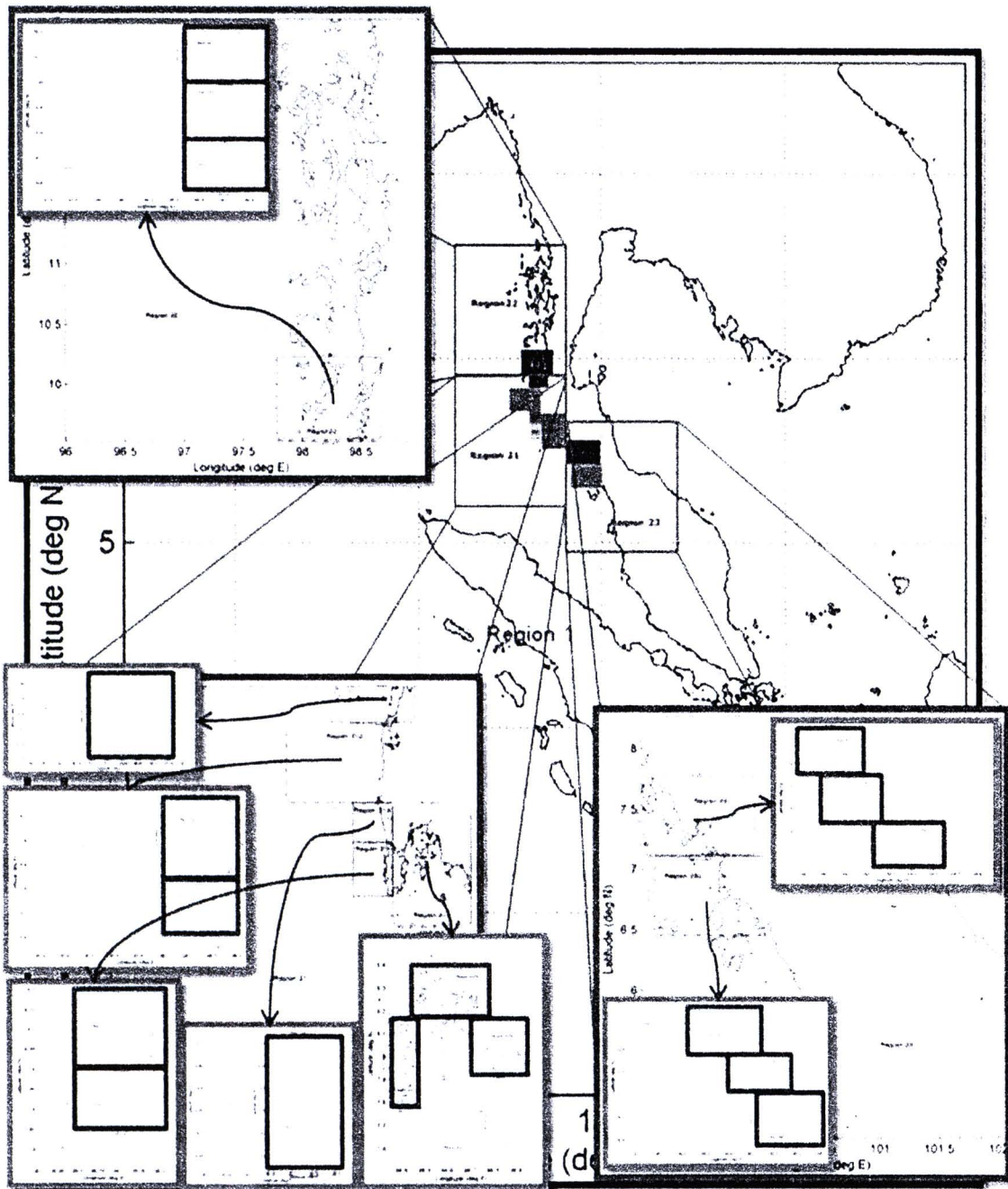
ระดับ ความ ละเอียด	หมายเลข	ความละเอียด		ขอบเขต												ขนาด ข้อมูลตาม แกน		ตำแหน่งบนระดับความ ละเอียดที่น้อยกว่า					
				ละติจูด						ลองจิจูด													
				จาก			ถึง			จาก			ถึง										
		'	"	°	'	"	°	'	"	°	'	"	°	'	"	X	Y	X	Y	X	Y		
R1	1	2	0	-10	0	0	18	0	0	87	0	0	110	0	0	690	840	-	-	-	-		
R2	21	0	15	5	59	45	9	30	0	95	59	45	99	0	0	721	841	271	481	380	585		
	22	0	15	9	31	45	13	2	0	95	59	45	99	0	0	721	841	271	587	380	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		



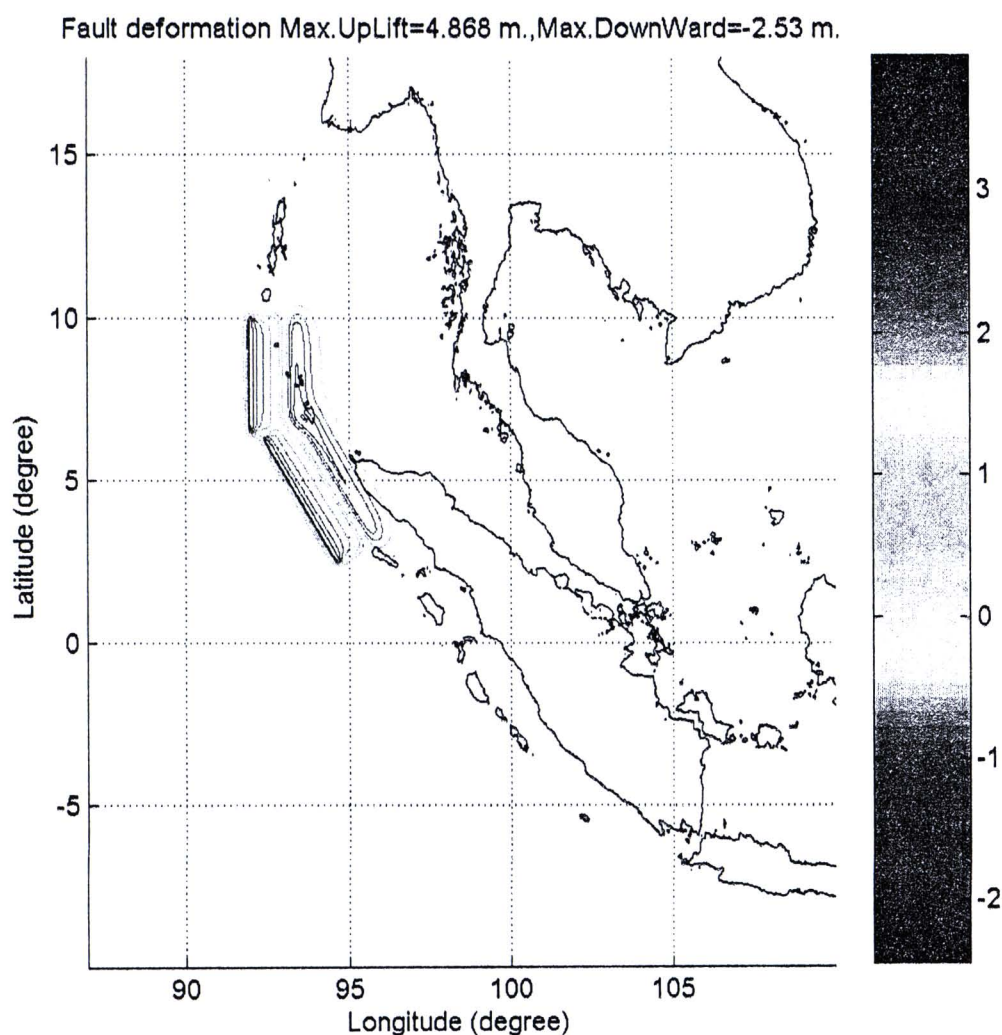
ส่วนข้อมูลที่ใช้ในการจำลองสึนามิ ได้มีการเก็บและสร้างข้อมูลจำลองสำหรับการคำนวณขึ้นมาจากข้อมูลที่สำรวจไว้จริง โดยมี ผศ.ดร.อาณัติ เรืองรัมย์ และนิสิตภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย เป็นผู้จัดทำข้อมูลส่วนนี้ขึ้นมาเป็นแฟ้มข้อมูล ซึ่งแฟ้มข้อมูลของระดับความสูงคลื่นผิวน้ำจะถูกเรียกว่า Deform และแฟ้มข้อมูลของระดับความลึกของน้ำทะเลเรียกว่า Region พร้อมทั้งระบุพิกัดและขนาดของพื้นที่ในแต่ละบริเวณ ความละเอียดที่ใช้ในการคำนวณในแต่ละบริเวณ ตำแหน่งของพื้นที่ภายในของแต่ละบริเวณ และระบุนี้อของพิกัดต่าง ๆ เป็นหมายเลขที่จะแสดงถึงระดับความละเอียดและลำดับของพื้นที่นั้น ๆ ที่จะใช้เป็นข้อมูลในการจำลองสึนามิตามตารางที่ 2-1 ที่แสดงข้อมูลของพิกัดพื้นที่ทั้งหมด 30 พื้นที่ ซึ่งแบ่งตามความละเอียดออกเป็น 4 ระดับ ที่มีพื้นที่ระดับ R4 เป็นพื้นที่ที่มีระดับความละเอียดมากที่สุดอยู่ 18 พื้นที่ ถูกครอบคลุมโดยพื้นที่ระดับ R3 ทั้งหมด 8 พื้นที่ และพื้นที่ระดับ R2 ทั้งหมด 3 พื้นที่ โดยทุกพื้นที่จะอยู่ซ้อน ๆ กันตามระดับความละเอียดภายในพื้นที่ระดับ R1 ซึ่งระดับความละเอียดจากพื้นที่ระดับ R1 ไป R2 เพิ่มขึ้น 8 เท่า จากพื้นที่ระดับ R2 ไป R3 และจากพื้นที่ระดับ R3 ไป R4 เพิ่มขึ้น 3 เท่า ตามรูปที่ 2-16 ที่แสดงความละเอียดที่เพิ่มขึ้นเมื่อเทียบกับพื้นที่ระดับ R1 ซึ่งข้อมูลนี้ได้แสดงให้เห็นค่าพิกัดในแนวเส้นละติจูดกับลองจิจูด และค่าพิกัดในการคำนวณตามแกนแนวนอน (X) กับแกนแนวตั้ง (Y) เพื่อใช้ระบุตำแหน่งการคำนวณของแต่ละพื้นที่ลงในโปรแกรม เมื่อนำข้อมูลมาพิกัดลงบนแผนที่จะได้ดังรูปที่ 2-17 แสดงให้เห็นการซ้อนกันของข้อมูลในแต่ละระดับความละเอียดที่ใช้เส้นกรอบแสดงขอบเขตของบริเวณต่าง ๆ ที่อยู่บนพื้นที่ที่มีระดับความละเอียดต่ำกว่า ส่วนรูปที่ 2-18 เป็นตัวอย่างข้อมูลระดับความสูงคลื่นผิวน้ำเริ่มต้นที่ใช้ในการคำนวณ ที่แสดงความสูงต่ำของคลื่นตามแถบสีทางด้านขวา



รูปที่ 2-16 ความละเอียดที่ต่างกันของแต่ละระดับ

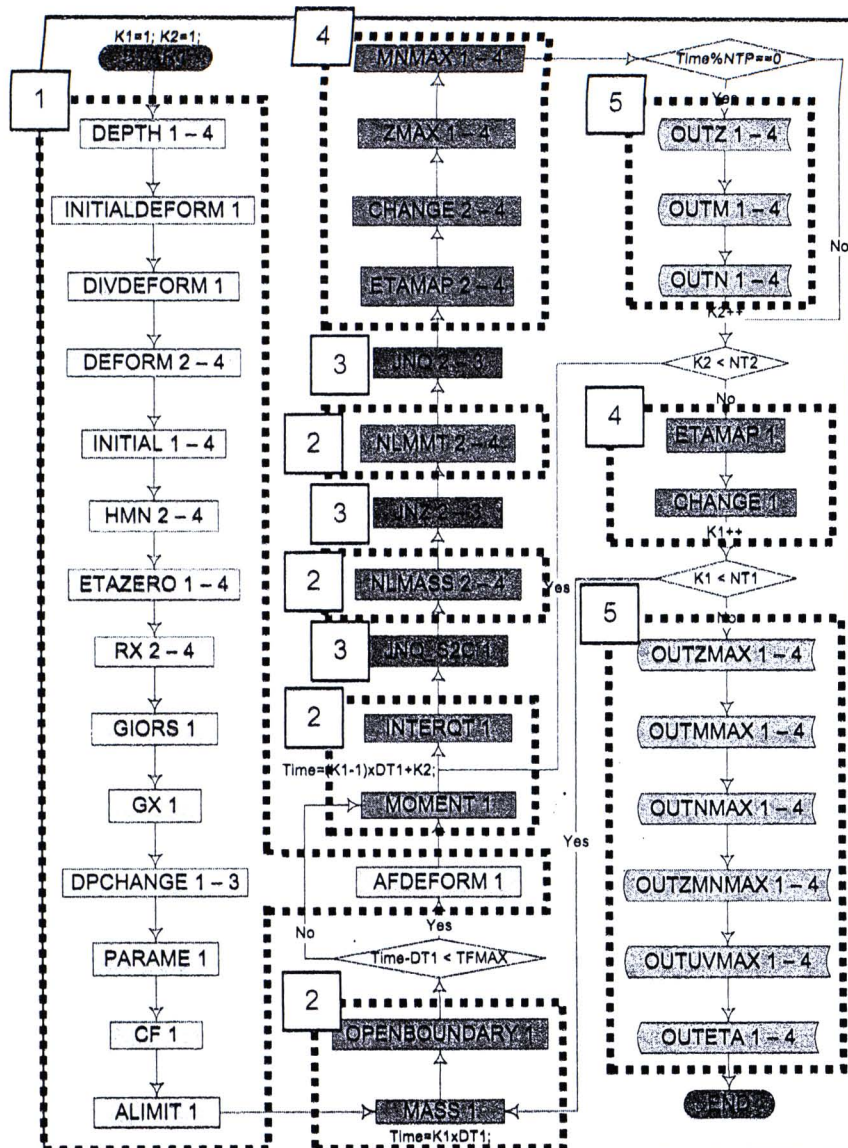


รูปที่ 2-17 พื้นที่ของบริเวณสนใจในการจำลองสึนามิ



รูปที่ 2-18 ตัวอย่างกรณีเกิดแผ่นดินไหวใต้มหาสมุทร

ในการจำลองสึนามิของโปรแกรมนี้ จะใช้หลักการอนุรักษ์มวลและโมเมนตัม (Conservation of Mass and Momentum) ทั้งการจำลองทั้งระยะใกล้และการจำลองระยะไกล โดยในการจำลองระยะใกล้จะคำนวณด้วยสมการเชิงเส้น (Linear Equations) ขณะที่การจำลองระยะไกลจะคำนวณด้วยสมการไม่เชิงเส้น (Nonlinear Equations) โปรแกรมจะทำการคำนวณด้วยหลักการอนุรักษ์มวลก่อนแล้วจึงคำนวณด้วยหลักการอนุรักษ์โมเมนตัม แต่ในระหว่างนั้นจะมีขั้นตอนของการแลกเปลี่ยนข้อมูลภายนอกในระดับความละเอียดแทรกอยู่ระหว่างการคำนวณ โดยมีลำดับการทำงานของโปรแกรมจะเป็นดังรูปที่ 2-19 ที่ใช้เส้นทึบแทนลำดับการทำงานของโปรแกรม และตัวเลข 1 – 4 ที่ระบุไว้แทนระดับความละเอียดทั้ง 4 ระดับที่จะถูกคำนวณในแต่ละลำดับการทำงาน



รูปที่ 2-19 ลำดับการคำนวณของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ

การทำงานของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ เมื่อวิเคราะห์ตามลักษณะการทำงานของโปรแกรมสามารถแบ่งออกได้เป็น 5 ส่วน ดังนี้

1. ส่วนที่หนึ่งแสดงถึงลำดับการทำงานที่ทำการกำหนดค่าตัวแปรต่าง ๆ เพื่อนำมาคำนวณ โดยในการทำงาน DEPTH เป็นการอ่านค่าระดับความลึกน้ำทะเล (HZ) จากแฟ้มข้อมูล region และในการทำงาน DIVDEFORM และ DEFORM เป็นการอ่านค่าเริ่มต้นของระดับความสูงคลื่นผิวน้ำ (Z) จากแฟ้มข้อมูล deform ส่วนการทำงาน INITIALDEFORM INITIAL HMN ETAZERO RX GIORs GX DPCHANGE PARAM CF ALIMIT และ AFDEFORM เป็นการกำหนดค่าตัว

แปรเริ่มต้นและค่าคงที่ต่าง ๆ ในการคำนวณ โดยในส่วนนี้ทำงานเพียงรอบเดียว เมื่อโปรแกรมเริ่มทำงานเท่านั้น

2. ส่วนที่สองแสดงถึงลำดับการทำงานที่ทำการคำนวณทิศทางการเคลื่อนที่ของคลื่น (M, N) และค่าระดับความสูงคลื่นผิวน้ำตามหลักการอนุรักษ์มวลและโมเมนตัม โดยมีการทำงาน MASS OPENBOUNDARY MOMENT และ INTERQT เป็นการคำนวณของพื้นที่ระดับ R1 ที่คำนวณด้วยสมการเชิงเส้นสำหรับการจำลองระยะไกล และการทำงาน NLMASS และ NLMMT เป็นการคำนวณของพื้นที่ระดับ R2 R3 และ R4 ที่คำนวณด้วยสมการไม่เชิงเส้นสำหรับการจำลองระยะใกล้
3. ส่วนที่สามแสดงถึงลำดับการทำงานที่ทำการแลกเปลี่ยนผลการคำนวณระหว่างพื้นที่ในบริเวณรอยต่อของแต่ละระดับความละเอียด โดยการทำงาน JNQ_S2C เป็นการส่งค่าระดับความสูงคลื่นผิวน้ำจากพื้นที่ระดับ R1 ให้กับพื้นที่ระดับ R2 ส่วนการทำงาน JNZ เป็นการส่งค่าระดับความสูงคลื่นผิวน้ำจากพื้นที่ระดับ R3 ให้กับพื้นที่ระดับ R2 และจากพื้นที่ระดับ R4 ให้กับพื้นที่ระดับ R3 และการทำงาน JNQ เป็นการส่งทิศทางการเคลื่อนที่ของคลื่นจากพื้นที่ระดับ R2 ให้กับพื้นที่ระดับ R3 และจากพื้นที่ระดับ R3 ให้กับพื้นที่ระดับ R4
4. ส่วนที่สี่แสดงถึงลำดับการทำงานที่ทำการเก็บผลการคำนวณต่าง ๆ ก่อนทำการบันทึกค่าลงแฟ้มข้อมูล หลังจากทำการคำนวณเสร็จเรียบร้อยแล้ว โดยมีการทำงาน ZMAX และ MNMAX เป็นการเก็บค่าสูงสุดของทิศทางการเคลื่อนที่ของคลื่นและระดับความสูงคลื่นผิวน้ำ การทำงาน ETAMAP เป็นการเก็บค่าเวลาที่คลื่นมาถึง ณ ตำแหน่งใด ๆ บนพื้นที่ และการทำงาน CHANGE เป็นการคัดลอกผลการคำนวณของเวลาปัจจุบัน เพื่อใช้เป็นค่าในการคำนวณของเวลาถัดไป
5. ส่วนที่ห้าแสดงถึงลำดับการทำงานที่ทำการบันทึกผลการคำนวณลงแฟ้มข้อมูล โดยมีการทำงาน OUTZ, OUTM และ OUTN เป็นการบันทึกผลของทิศทางการเคลื่อนที่ของคลื่นและค่าระดับความสูงคลื่นผิวน้ำลงแฟ้มข้อมูลตามช่วงเวลาที่กำหนด ซึ่งค่าระดับความสูงคลื่นผิวน้ำจะถูกนำมาสร้างเป็นภาพจำลองการเกิดคลื่นสึนามิที่ใช้ในการเตือนภัย และการทำงาน OUTZMAX, OUTMMAX, OUTNMAX, OUTZMNMAX, OUTUVMAX และ OUTETA เป็นการบันทึกผลการคำนวณสูงสุดของค่าตัวแปรต่าง ๆ ลงแฟ้มข้อมูล

โดยในการทำงานของโปรแกรมนี้ ได้แบ่งช่วงเวลาที่ทำการจำลองออกเป็นขั้น ๆ (Timesteps) เพื่อทำการคำนวณค่าของพื้นที่ระดับ R1 ทุก ๆ 4 วินาที และคำนวณค่าของพื้นที่ระดับ R2, R3 และ R4 ทุก ๆ วินาที ซึ่งจะทำการจำลองเหตุการณ์เป็นเวลา 36,000 วินาที หรือ 10 ชั่วโมงนับจากเวลาที่เกิดแผ่นดินไหว และทำการบันทึกข้อมูลที่ใช้ในการสร้างภาพจำลองสึนามิ ทุก ๆ 60 วินาที หรือ 1 นาที ซึ่งค่าของเวลาต่าง ๆ นี้ สามารถปรับเปลี่ยนได้จากตัวแปรในโปรแกรม โดยค่าเวลาต่าง ๆ ที่ได้กำหนดมานี้ เป็นค่าเวลาที่ทางภาควิชาวิศวกรรมโยธา จุฬาลงกรณ์มหาวิทยาลัย ได้กำหนดขึ้นมาได้ตามความเหมาะสมในการจำลองสึนามิเพื่อการเตือนภัย แต่เนื่องจากการเตือนภัยสึนามินั้น นอกจากต้องมีความถูกต้องแม่นยำแล้ว ยังต้องการประสิทธิภาพในการคำนวณที่รวดเร็ว ซึ่งในการจำลองสึนามิเพื่อเตือนภัยนั้นควรใช้เวลาไม่เกินห้าสิบนาที ดังนั้นจึงจำเป็นต้องหาวิธีการที่จะทำให้การจำลองสึนามิด้วยการคำนวณแบบลำดับ สามารถคำนวณได้โดยใช้เวลาในการคำนวณที่ลดลงเป็นอย่างมาก

2.2.2 โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนาน (Parallel Tsunami Simulation Program)

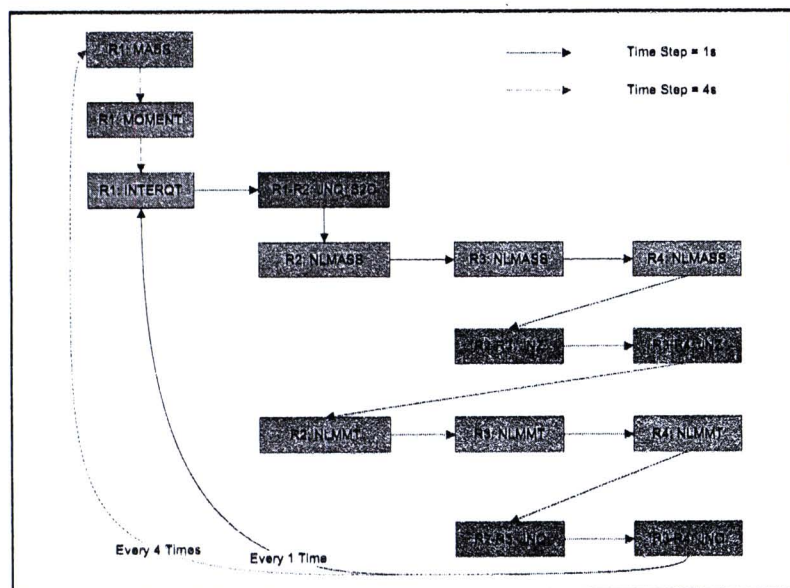
เนื่องจากโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ ใช้เวลาประมาณ 4 – 6 ชั่วโมงต่อการคำนวณหนึ่งโซน ถ้าต้องคำนวณการเกิดสึนามิด้วยพื้นที่ทั้งหมด 18 โซน ด้วยกรณีของการเกิดแผ่นดินไหวมากกว่า 1,000 กรณี ก็ต้องใช้เวลามากกว่า 18,000 เท่าบนเครื่องคอมพิวเตอร์เครื่องเดียว และได้ผลการคำนวณที่ซ้ำซ้อนกันในหนึ่งกรณี จึงได้นำโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับนี้มาพัฒนาให้เป็นการคำนวณแบบขนาน ที่ช่วยให้มีประสิทธิภาพและความเร็วในการจำลองสึนามิเพิ่มขึ้น

การคำนวณการเกิดคลื่นสึนามิของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานนี้ เป็นการพัฒนาและปรับปรุงการคำนวณจากโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับที่ใช้ Message Passing Interface (MPI Library) มาช่วยในการคำนวณแบบขนาน ซึ่งเป็นโครงงานทางวิศวกรรมเรื่อง “โปรแกรมแบบขนานเพื่อจำลองการเกิดคลื่นสึนามิ” [14] ของนายกิตติพัฒน์ วิโรจน์ศิริ นิสิตภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี 2550 โดยมี ผศ. ดร. วีระ เหมืองสิน เป็นอาจารย์ที่ปรึกษาโครงงาน เพื่อให้ประสิทธิภาพของโปรแกรมจำลองสึนามิดีขึ้น และให้สามารถคำนวณได้แม่นยำเหมือนโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับเดิม แต่ใช้ระยะเวลาในการคำนวณที่ลดลง เพื่อเป็นการใช้ประโยชน์จากทรัพยากรคอมพิวเตอร์ที่มีอยู่อย่างสูงสุด ในการทำงานบนระบบคลัสเตอร์คอมพิวเตอร์ (Cluster computer) ซึ่งตัวโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานนี้ ได้ทำการทดสอบบนระบบคลัสเตอร์ TERA ของ Thai National Grid Center ในมหาวิทยาลัยเกษตรศาสตร์ ประเทศไทย และระบบคลัสเตอร์

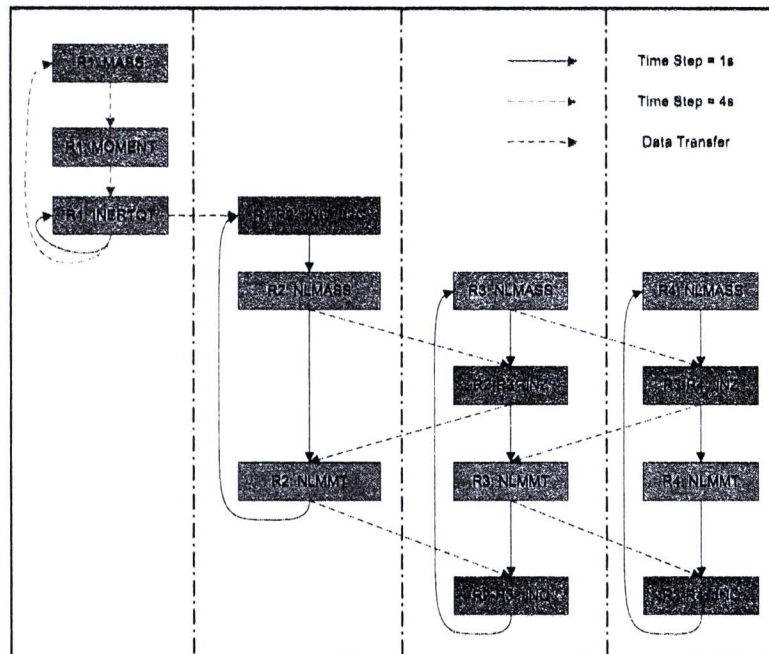


TSUBAME ของ GSIC Center ของสถาบันเทคโนโลยีแห่งโตเกียว (Tokyo Institute of Technology หรือ TiTech) ประเทศญี่ปุ่น โปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานจะใช้ส่วนการคำนวณจากโปรแกรมจำลองสึนามิด้วยการคำนวณแบบลำดับ ที่เขียนด้วยชุดคำสั่งภาษา FORTRAN แต่การเรียกใช้งานส่วนการคำนวณและการควบคุมการทำงานแบบขนานด้วย MPI เขียนด้วยชุดคำสั่งภาษา C++

แนวทางในการแก้ปัญหาด้วยการคำนวณแบบขนานนี้ ประกอบด้วยสองขั้นตอน โดยขั้นแรกเป็นการแบ่งการคำนวณออกตามพื้นที่ที่ระดับความละเอียดต่าง ๆ และให้แต่ละหน่วยประมวลผลรับผิดชอบการคำนวณในหนึ่งระดับความละเอียด ซึ่งการแบ่งการคำนวณแบบขนานจำเป็นต้องคำนึงถึงการเชื่อมต่อของข้อมูล หรือการใช้งานข้อมูลร่วมกันภายนอกระดับความละเอียดดังรูปที่ 2-20 ที่แสดงให้เห็นลำดับการทำงานของ การคำนวณแบบลำดับที่แยกย่อยให้เห็นถึงลำดับการทำงานในส่วนที่มีการเชื่อมต่อกันของข้อมูลภายนอกระดับความละเอียด และจากรูปที่ 2-21 ที่แสดงถึงลำดับการทำงานด้วยการคำนวณแบบขนานที่จะแยกกันทำงานด้วยแต่ละหน่วยประมวลผล ที่ใช้เส้นยาวสลับสั้นเป็นแนวแบ่งการทำงานของแต่ละหน่วยประมวลผล ใช้เส้นทึบสีเข้มแทนลำดับการทำงานที่จะทำทุก ๆ รอบการทำงาน ใช้เส้นประสีอ่อนแทนลำดับการทำงานที่ทำทุก ๆ 4 รอบการทำงาน และใช้เส้นประสีเข้มแทนการส่งผ่านข้อมูลระหว่างหน่วยประมวลผล ซึ่งจากลักษณะของการคำนวณที่แยกออกตามระดับความละเอียดของพื้นที่ที่เก็บสมการ และมีการส่งผ่านข้อมูลระหว่างกันเฉพาะรอยต่อ จึงสามารถทำการแบ่งการคำนวณตามระดับความละเอียดได้สำหรับการคำนวณแบบขนาน

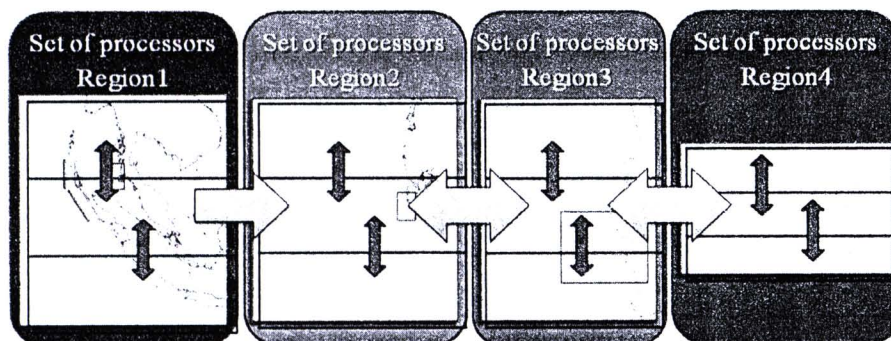


รูปที่ 2-20 ลำดับการทำงานของ การคำนวณแบบลำดับในส่วนการคำนวณ



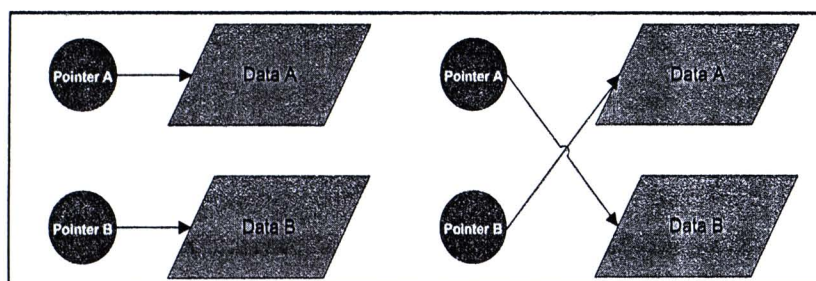
รูปที่ 2-21 ลำดับการทำงานเมื่อพัฒนาโปรแกรมเป็นการคำนวณแบบขนาน

ส่วนขั้นที่สองเป็นการแบ่งการคำนวณในพื้นที่เดียวกันออกเป็นหลาย ๆ งาน เพื่อให้แต่ละหน่วยประมวลผลรับผิดชอบในการคำนวณ ซึ่งการแบ่งการคำนวณแบบนี้ช่วยลดขนาดของพื้นที่ในการคำนวณต่อหนึ่งหน่วยประมวลผลที่ทำให้ระยะเวลาการคำนวณลดลงด้วย แต่ด้วยความยากในการแบ่งงานให้ได้ประสิทธิภาพและความเร็วอันเนื่องมาจากความซับซ้อนของปัญหาที่มีหลายระดับความละเอียด จึงต้องทำการเก็บข้อมูลการทำงานของโปรแกรมโดยละเอียดเพิ่มเติม นั่นคือเมื่อทำตามสองขั้นตอนแล้วได้เป็นการแบ่งงานตามรูปที่ 2-22



รูปที่ 2-22 การแบ่งงานของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนาน

นอกจากนั้นการพัฒนาโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานนี้ ได้ใช้วิธี Page Flipping ดังรูปที่ 2-23 ที่เป็นวิธีการสลับตัวแปรที่ชี้ตำแหน่งของข้อมูลในตัวแปรแทนการเขียนทับข้อมูลในตัวแปรทั้งหมด มาทำงานแทนการทำงาน CHANGE ซึ่งทำให้เวลาที่ใช้ลดลงจาก $O(mn)$ เป็น $O(1)$



รูปที่ 2-23 ตัวอย่างการทำ Page Flipping

จากการวิเคราะห์ความซับซ้อนเชิงเวลา (Time complexity) ในการคำนวณของแต่ละส่วนย่อยของโปรแกรมที่มีการทำงานซ้ำ ๆ ตลอดการทำงาน ซึ่งเป็นส่วนการคำนวณหลักของโปรแกรม โดยที่กำหนดให้ m คือความยาวของข้อมูลตามแกน X และ n คือความยาวของข้อมูลตามแนวแกน Y ได้เป็นค่าบิกโอ (Big O notation) ดังนี้

1. ส่วนที่ทำการคำนวณตามหลักการอนุรักษ์มวลและโมเมนตัม ซึ่งจะประกอบด้วยส่วนการคำนวณที่ใช้ในการจำลองระยะไกล และการจำลองระยะใกล้
 - MASS ใช้ระยะเวลาในการทำงาน $O(mn)$
 - OPENBOUNDARY ใช้ระยะเวลาในการทำงาน $O(m+n)$
 - MOMENT ใช้ระยะเวลาในการทำงาน $O(mn)$
 - INTERQT ใช้ระยะเวลาในการทำงาน $O(mn)$
 - NLMASST ใช้ระยะเวลาในการทำงาน $O(mn)$
 - NLMMT แบ่งการทำงานออกเป็นสองขั้นตอนคือ NLMMT1 และ NLMMT2 ใช้ระยะเวลาในการทำงานส่วนการคำนวณละ $O(mn)$
2. ส่วนที่ทำการเชื่อมต่อกันของข้อมูลภายนอกในระดับความละเอียด
 - JNQ_S2C ใช้ระยะเวลาในการทำงาน $O(m+n)$
 - JNZ ใช้ระยะเวลาในการทำงาน $O(m+n)$
 - JNQ ใช้ระยะเวลาในการทำงาน $O(m+n)$

3. ส่วนที่ทำการจัดการผลการคำนวณ

- CHANGE ใช้ระยะเวลาในการทำงาน $O(mn)$
- ZMAX และ MNMAX ใช้ระยะเวลาในการทำงาน $O(mn)$
- OUTZ OUTM และ OUTN ใช้ระยะเวลาในการทำงาน $O(mn)$

เมื่อทำการวัดระยะเวลาที่ใช้ในการทำงานของแต่ละส่วนการคำนวณในแต่ละรอบการทำงาน (Timestep) ด้วยพื้นที่โซน 2111 และโซน 2121 ได้ผลการทำงานตามตารางที่ 2-2 โดยแสดงระยะเวลาที่ใช้ในการทำงานของส่วนการคำนวณที่มีการเรียกใช้งานซ้ำ ๆ ตลอดการทำงานจากการจำลองสึนามิด้วยการคำนวณแบบลำดับของพื้นที่โซน 2111 และโซน 2121 ที่แสดงให้เห็นอย่างชัดเจนว่า แต่ละส่วนการคำนวณใช้เวลาในการทำงานมาน้อยเพียงใด โดยส่วนการคำนวณ ZMAX และ MNMAX แทนด้วยส่วนการทำงาน MAX ส่วนการคำนวณ OUTZ OUTM และ OUTN แทนด้วยส่วนการทำงาน OUTB และส่วนการคำนวณ OUTZMAX, OUTMMAX, OUTNMAX, OUTZMNMAX, OUTUVMAX และ OUTETA แทนด้วยส่วนการทำงาน OUTMAX

โดยส่วนการคำนวณของโปรแกรมนี้ประกอบด้วย ส่วนที่ทำการคำนวณตามหลักการอนุรักษ์มวลและโมเมนตัม และส่วนที่ทำการเชื่อมต่อกันของข้อมูลภายนอกในระดับความละเอียด จะทำการคำนวณทั้งหมด 36,000 รอบ ในระดับพื้นที่ระดับ R2 R3 และ R4 แต่ในระดับความละเอียดพื้นที่ระดับ R1 ทำการคำนวณทั้งหมด 9,000 รอบ และส่วนที่ทำการจัดการผลการคำนวณที่ต้องมีการเขียนข้อมูลลงแฟ้มข้อมูลของทั้ง 4 ระดับความละเอียด ซึ่งต้องทำงานทั้งหมด 600 รอบ และส่วนกำหนดค่าเบื้องต้นและส่วนที่ทำการเขียนผลลัพธ์สูงสุดต่าง ๆ ของการคำนวณทำงานเพียงหนึ่งรอบ เมื่อรวมระยะเวลาที่ต้องใช้ในการทำงานทั้งหมดแล้ว จึงทำให้ต้องใช้เวลานานในการคำนวณแต่ละครั้งที่ทำการจำลองสึนามิเป็นเวลานานมากตามระยะเวลาที่ต้องการจำลองและขนาดของพื้นที่ในแต่ละระดับความละเอียด

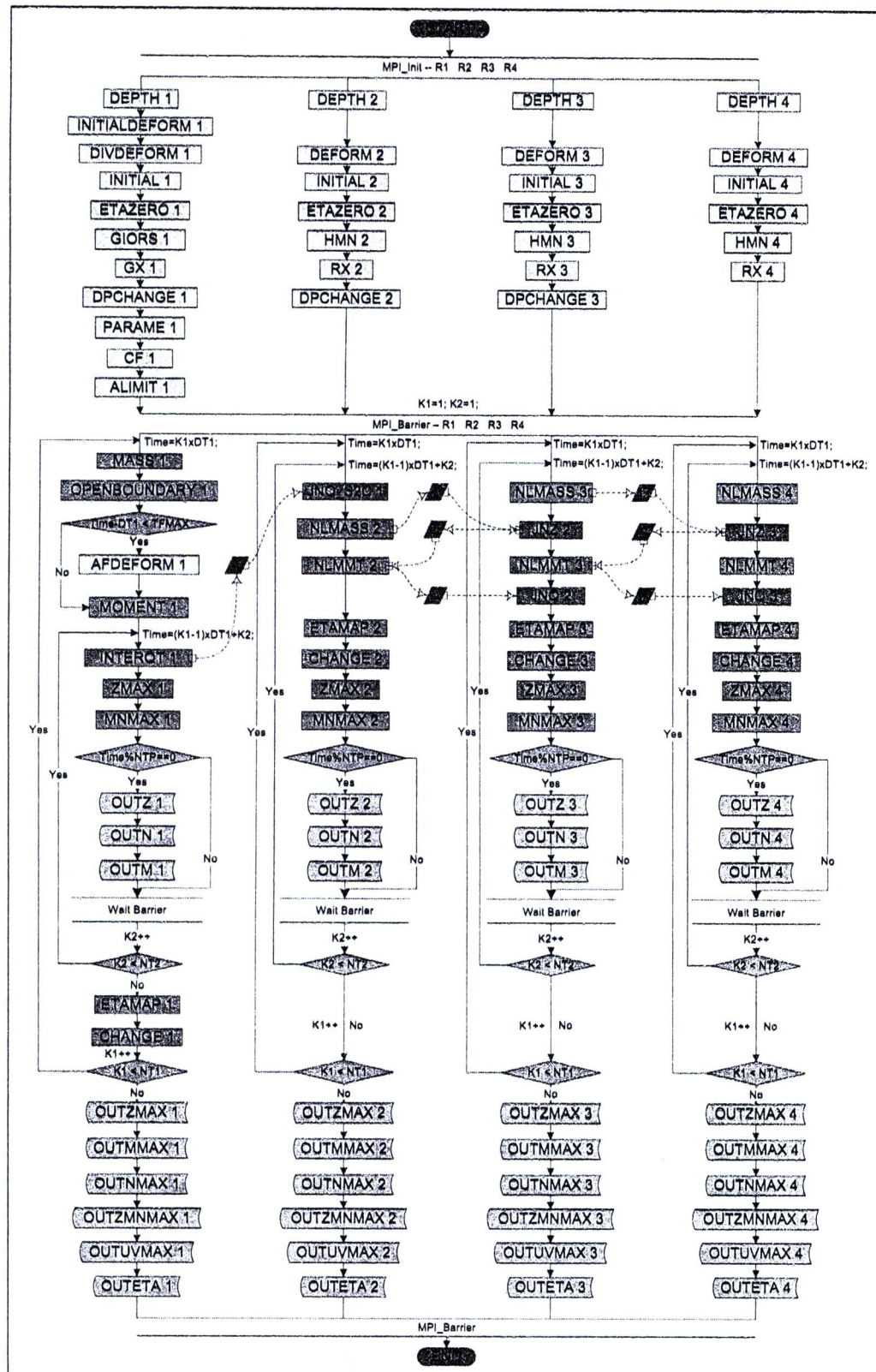
ตารางที่ 2-2 ระยะเวลาที่ใช้ในการทำงานของส่วนการคำนวณที่มีการเรียกใช้งานซ้ำ ๆ ตลอดการ
ทำงานจากการจำลองสึนามิตัวด้วยการคำนวณแบบลำดับของพื้นที่โซน 2111 และโซน 2121

พื้นที่	ฟังก์ชัน	จำนวนรอบ	ระยะเวลาที่ใช้ในการทำงาน (วินาที)			
			โซน 2111		โซน 2121	
			เวลารวม	เวลาเฉลี่ย	เวลารวม	เวลาเฉลี่ย
R1-R4	INITIAL	1	1.330	1.330	1.700	1.700
R1	MASS	9000	62.259	0.006918	58.835	0.006537
R1	OPENBOUNDARY	9000	2.046	0.000227	2.189	0.000243
R1	MOMENT	9000	139.495	0.015499	137.773	0.015308
R1	INTERQT	36000	504.199	0.014006	483.037	0.013418
R1	JNQ_S2C	36000	10.380	0.000288	10.129	0.000281
R2	NLMASS	36000	297.165	0.008255	291.829	0.008106
R3	NLMASS	36000	36.863	0.001024	140.067	0.003891
R4	NLMASS	36000	77.268	0.002146	145.040	0.004029
R2	JNZ	36000	5.054	0.000140	8.221	0.000228
R3	JNZ	36000	3.592	0.000100	6.205	0.000172
R2	NLMMT	36000	7,344.722	0.204020	7,218.833	0.200523
R3	NLMMT	36000	891.235	0.024757	3,601.958	0.100054
R4	NLMMT	36000	1,764.622	0.049017	3,310.396	0.091955
R2	JNQ	36000	3.983	0.000111	6.055	0.000168
R3	JNQ	36000	3.551	0.000099	4.722	0.000131
R2	CHANGE	36000	407.279	0.011313	400.067	0.011113
R3	CHANGE	36000	52.432	0.001456	200.602	0.005572
R4	CHANGE	36000	107.533	0.002987	203.914	0.005664
R1	MAX	36000	572.768	0.015910	573.358	0.015927
R2	MAX	36000	611.812	0.016995	611.684	0.016991
R3	MAX	36000	80.343	0.002232	298.889	0.008302
R4	MAX	36000	160.317	0.004453	297.979	0.008277
R1	OUTB	600	373.477	0.622462	372.594	0.620990
R2	OUTB	600	388.917	0.648195	386.988	0.644980
R3	OUTB	600	55.964	0.093273	188.922	0.314870
R4	OUTB	600	102.707	0.171178	187.418	0.312363
R1	CHANGE	9000	97.575	0.010842	95.323	0.010591
R1-R4	OUTMAX	1	86.860	86.860	107.561	107.561
TOTAL		1	14,250.710	14,250.710	19,356.971	19,356.971

เมื่อวิเคราะห์ส่วนการคำนวณของลำดับการทำงานที่สามารถนำมาปรับปรุงเป็นการคำนวณแบบขนานได้แล้ว ลำดับการทำงานทั้งหมดของการคำนวณแบบขนานจะเป็นดังรูปที่ 2-24 ที่ใช้เส้นทึบแทนลำดับการทำงานของโปรแกรม ใช้เส้นประแทนการส่งข้อมูลภายนอกกระดานความละเอียด และตัวเลข 1 - 4 แทนระดับความละเอียดทั้ง 4 ระดับ ซึ่งรายละเอียดและลำดับการทำงานของโปรแกรมเป็นดังนี้คือ

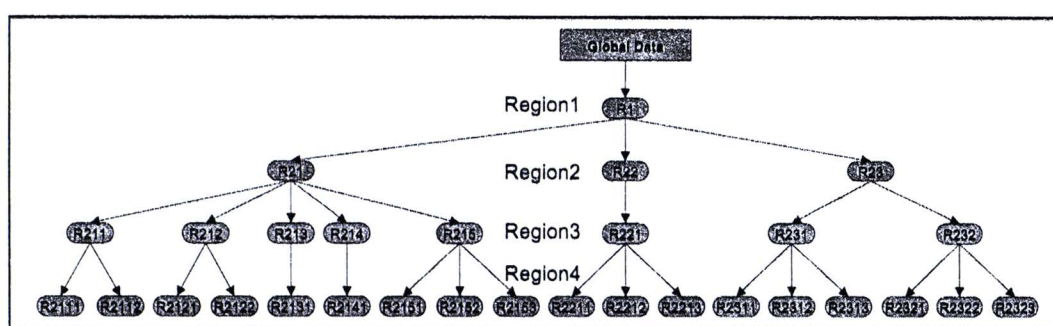
1. ก่อนที่โปรแกรมจะเริ่มต้นการทำงาน จะใช้โปรแกรมแบ่งการคำนวณ (Splitter) เพื่อกำหนดขอบเขตของข้อมูลในแต่ละหน่วยประมวลผลต้องรับผิดชอบตามจำนวนงานที่แบ่งในแต่ละระดับความละเอียด พร้อมทั้งระบุหมายเลขของแต่ละขอบเขตด้วยจากแฟ้มโครงแบบข้อมูลของโปรแกรม
2. เมื่อเริ่มต้นเข้าสู่การทำงานของโปรแกรม แต่ละหน่วยประมวลผลจะทำการตรวจสอบหมายเลขของงานที่รับผิดชอบให้ตรงกับหมายเลขตามขอบเขตของข้อมูลจากแฟ้มโครงแบบข้อมูลของโปรแกรม ว่าต้องคำนวณในระดับความละเอียดใด และที่ตำแหน่งใดบนระดับความละเอียดนั้น
3. จากนั้นแต่ละหน่วยประมวลผลเริ่มทำงานในส่วนการจัดการค่าเริ่มต้นในการคำนวณของแต่ละหน่วยประมวลผลได้ทันที เนื่องจากในส่วนนี้ไม่มีการเชื่อมต่อกันของข้อมูลระหว่างหน่วยประมวลผล
4. เมื่อจะเข้าสู่ลำดับการทำงานในส่วนการคำนวณหลัก จะให้ทุกหน่วยประมวลผลหยุดการทำงานก่อนเพื่อรอให้ทำงานในส่วนนี้ไปพร้อม ๆ กัน เพราะระหว่างการทำงานจะมีการเชื่อมต่อและส่งผ่านข้อมูลกันระหว่างหน่วยประมวลผล ที่จะมีผลต่อผลลัพธ์ในการคำนวณ และมีการรอให้ทำงานพร้อม ๆ กันในการคำนวณของแต่ละหน่วยประมวลผลเมื่อทำงานเสร็จในทุก ๆ รอบการคำนวณ
5. ก่อนจะสิ้นสุดการทำงาน ส่วนคำนวณในทุกหน่วยประมวลผลก็จะหยุดการทำงาน เพื่อรอที่จะสิ้นสุดการทำงานของโปรแกรมพร้อม ๆ กัน เพราะก่อนจบโปรแกรม จะมีการลบค่าตัวแปรต่าง ๆ ที่ใช้ในการคำนวณทั้งหมด





รูปที่ 2-24 ลำดับการทำงานของโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนาน

โดยแฟ้มโครงแบบข้อมูลของโปรแกรม (Config.xml) เป็นแฟ้มข้อมูลที่บันทึกโครงแบบข้อมูลของพื้นที่ในการคำนวณด้วยรูปแบบแฟ้มข้อมูล XML (eXtensible Markup Language) [15] ซึ่งเป็นรูปแบบแฟ้มข้อมูลที่มนุษย์สามารถเข้าใจได้ง่าย และมีคลัง (Library) สำหรับจัดการแฟ้มข้อมูลที่แพร่หลาย โดยใช้โครงสร้างข้อมูลแบบต้นไม้ดังรูปที่ 2-25 ที่มีรากของต้นไม้ เป็นข้อมูลหลักของตัวแปรต่าง ๆ ที่ใช้ในการคำนวณ และมีลูกของรากเป็นข้อมูลของพื้นที่ระดับ R1 ที่จะมีลูกเป็นพื้นที่ระดับ R2 R3 และ R4 ที่มีระดับความละเอียดที่เพิ่มขึ้น ตามลำดับ ซึ่งจะกำหนดพื้นที่การคำนวณของแต่ละหน่วยประมวลผล กำหนดจำนวนการแบ่งการคำนวณในแต่ละพื้นที่ และการกำหนดชื่อแฟ้มข้อมูลที่เกี่ยวข้องในการคำนวณ



รูปที่ 2-25 รูปแบบของโครงสร้างข้อมูลแบบต้นไม้

ในการทดสอบโปรแกรมจำลองสึนามิด้วยการคำนวณแบบขนานที่เคยมีการทดสอบมาก่อนหน้านี้ได้แบ่งการทดสอบออกเป็น 4 ส่วน คือ

1. การทดสอบผลกระทบที่มีต่อประสิทธิภาพของการจัดแบ่งงาน – ในการทดสอบนี้จะทำการกำหนดจำนวนของงานที่แบ่งให้คงที่ และทำการปรับเปลี่ยนในส่วนของจำนวนการแบ่งตามคอรัลล์ และตามแถว โดยรูปแบบของการจัดการงานและการแบ่งงานเพื่อทดสอบประสิทธิภาพของโปรแกรม ได้ใช้การทดสอบกับรูปแบบของปัญหาเพียงชนิดเดียว และทำการแบ่งงานโดยให้พื้นที่ระดับ R1 แบ่งเป็น 2 ส่วนตามแถวเท่านั้น ส่วนพื้นที่ระดับ R2 R3 และ R4 แบ่งพื้นที่โดยใช้ค่า 40,000 เป็นตัวหารขนาดของพื้นที่ให้ได้เป็นจำนวนงานที่แบ่ง
2. การทดสอบผลกระทบที่มีต่อประสิทธิภาพของวิธีการสั่งงาน – ในการทดสอบนี้จะทำการกำหนดจำนวนของงานที่แบ่งให้คงที่ และทำการเปรียบเทียบการทำงานของโปรแกรมที่มีการทำงานซ้อนทับกันของการคำนวณและการสื่อสาร (Overlapping) และโปรแกรมที่ไม่มีการทำงานซ้อนทับกันของการคำนวณและ

การสื่อสาร (Non-overlapping) โดยใช้รูปแบบของการจัดการงานและการแบ่งงานเพื่อทดสอบประสิทธิภาพของโปรแกรมเหมือนเดิม

3. การทดสอบผลกระทบที่มีต่อประสิทธิภาพของระบบที่แตกต่างกัน – ในการทดสอบนี้ จะทำการกำหนดจำนวนของงานที่แบ่งให้คงที่ และทำการทดสอบในระบบที่แตกต่างกัน ซึ่งจะทดสอบด้วยระบบคลัสเตอร์ TERA เทียบกับระบบคลัสเตอร์ TSUBAME โดยใช้รูปแบบของการจัดการงานและการแบ่งงานเพื่อทดสอบประสิทธิภาพของโปรแกรมเหมือนเดิม
4. การทดสอบความสามารถในการขยายขนาดของปัญหา – ในการทดสอบนี้ทำการเปรียบเทียบการทำงานของโปรแกรมเมื่อมีจำนวนของงานที่แบ่ง และจำนวนการแบ่งตามคอลัมน์ และตามแถวที่แตกต่างกัน โดยมีรูปแบบของปัญหาเพียงโซนเดียว และไม่คำนึงถึงรูปแบบของการแบ่งงาน

ผลการทดสอบจากการพัฒนาและปรับปรุงโปรแกรมจำลองสินามิที่ได้นั้น ในเรื่องของความเร็วในการคำนวณ สามารถทำได้อย่างมีประสิทธิภาพเป็นที่น่าพอใจ แต่ยังไม่มียุทธวิธีในการจัดการงานเพื่อหารูปแบบในการแบ่งงานในการคำนวณแบบขนานที่แน่นอนและชัดเจน โดยในการคำนวณแบบขนานนี้มีความซับซ้อนของข้อมูลที่น่ามาคำนวณในแต่ละระดับความละเอียด ซึ่งข้อมูลที่น่ามาคำนวณได้ถูกแบ่งตามระดับความละเอียดของข้อมูล มีความละเอียดและขนาดของพื้นที่ในแต่ละระดับความละเอียดที่แตกต่างกัน มีจำนวนรอบในการทำงานที่แตกต่างกันของการจำลองระยะใกล้กับการจำลองระยะไกล และสามารถจัดการแบ่งงานของแต่ละระดับความละเอียดได้ อีกทั้งยังเพิ่มจำนวนพื้นที่ในแต่ละระดับความละเอียดที่จะให้ทำการคำนวณในแต่ละครั้งได้ด้วย ด้วยเหตุนี้ จึงจำเป็นต้องหารูปแบบของการแบ่งงานและการกระจายงานที่เหมาะสมในการคำนวณแบบขนาน เพื่อสร้างเป็นระเบียบวิธีการจัดการงานมาตรฐานที่มีประสิทธิภาพในการคำนวณ