สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โครงการวิจัยปีงบประมาณ 2547

เรื่อง

โครงการเครื่องให้บริการประมวลผลขนานความเร็วสูง แบบคลัสเตอร์

RCH
QA
76.58
.A756C
เลขหมู่.....
เลขทะเบียน 79727
วัน,เดือน,ปี 11 เม.ย. 2551

จัดทำโดย ฝ่ายควบคุมเครื่อง
สำนักวิจัยและบริการคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b 11903434
i

สารบัญ

บทคัดย่อ 4

Abstract 4

บทที่ *1* โครงสร้างของคอมพิวเตอร์ขนานแบบคลัสเตอร์ 5

1.1 บทนำ 5

1.2 ความสำคัญของคอมพิวเตอร์ความเร็วสูง 5

1.3 ชนิดและโครงสร้างของคอมพิวเตอร์แบบขนาน 6

1.3.1 ระบบแบบหลายหน่วยประมวลผล (Multiprocessor) 7

1.3.2 ระบบแบบหลายเครื่องประมวลผล (Multi computer) 8

1.4 ส่วนประกอบโดยรวมของคอมพิวเตอร์แบบคลัสเตอร์ 12

1.5 การจำแนกคอมพิวเตอร์แบบคลัสเตอร์ 14

1.5.1 จำแนกตามงานที่ประยุกต์ใช้ 14

1.5.2 จำแนกตามลักษณะของเครื่องในระบบ 14

1.6 โครงสร้างพื้นฐานของระบบคลัสเตอร์ 15

1.6.1 ส่วนประกอบทางฮาร์ดแวร์ 15

1.6.2 ระบบปฏิบัติการ 16

1.6.3 คลัสเตอร์มิดเดิลแวร์ 16

1.6.4 การเชื่อมต่อเครือข่าย 18

1.7 เครื่องมือและโปรแกรมอัตถะประโยชน์ 22

1.7.1 โปรแกรมสื่อสารระหว่างโหนดโดยการส่งข้อความ 22

1.7.2 คอมไพเลอร์ 23

1.7.3 ชุดคำสั่งทางคณิตศาสตร์ 23

1.8 การจัดการระบบคลัสเตอร์ 24

1.9 สรุป 25

บทที่ *2* โปรแกรมแบบขนาน 26

2.1 บทนำ 26

2.2 รูปแบบของการพัฒนาโปรแกรมแบบขนาน 26

2.2.1 การสร้างโปรแกรมแบบขนานโดยอัตโนมัติ 26

2.2.2 การสร้างโปรแกรมโดยใช้ชุดคำสั่งแบบขนาน 27

2.2.3 การสร้างโปรแกรมแบบขนานด้วยตัวเอง 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 กลวิธีการโปรแกรมแบบขนาน 28
2.4 การออกแบบโปรแกรมขนาน 31
2.4.1 ขั้นตอนการแบ่งงาน 32
2.4.2 ขั้นตอนออกแบบการสื่อสาร 33
2.4.3 ขั้นตอนการรวมกลุ่มงาน 35
2.4.4 ขั้นตอนการกำหนดงาน 36
2.5 การสร้างภาพเชิงปริมาตรแบบฉายแสงบนระบบคลัสเตอร์ 36
2.5.1 การแบ่งงานในการสร้างภาพเชิงปริมาตร 37
2.5.2 การรวมกลุ่มงาน 37
2.5.3 การสื่อสารระหว่างงานในการสร้างภาพเชิงปริมาตร 38
2.5.4 การกำหนดงานในการสร้างภาพเชิงปริมาตร 38
2.6 สรุป 38
บทที่ 3 38
สรุปผลการวิจัย 39
3.1 โครงสร้างทาง Hardware 39
3.2 โครงสร้างทาง Software 39
3.3 รายการวัสดุอุปกรณ์ 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง โครงการเครื่องให้บริการประมวลผลขนานความเร็วสูงแบบคลัสเตอร์
(High Performance Clustering Server)

โดย นายนพรัตน์ พันธุ์เสนา หัวหน้าโครงการวิจัย
นายสมโชค กิ้มปาน
รตท. สมชาติ เลิกบางพลัด
สำนักวิจัยและบริการคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

เนื่องจากแนวโน้มของงานวิจัยหลาย ๆ อย่างในปัจจุบันของสถาบันฯ มีความจำเป็นต้องใช้ความสามารถในการประมวลผลของคอมพิวเตอร์ที่มีประสิทธิภาพสูงมาก เพื่อให้การคำนวณในงานวิจัยสามารถได้ผลลัพธ์ที่ถูกต้องและรวดเร็ว แต่เนื่องจากคอมพิวเตอร์ประมวลผลความเร็วสูงหรือซุปเปอร์คอมพิวเตอร์มีราคาแพงมาก ดังนั้นเพื่อเป็นการทดแทนเทคโนโลยีราคาแพง และเพื่อเป็นการตอบสนองต่อความต้องการในงานวิจัยเหล่านี้ ทางสำนักวิจัยและบริการคอมพิวเตอร์จึงได้จัดทำโครงการเครื่องให้บริการประมวลผลขนานความเร็วสูงแบบคลัสเตอร์ขึ้นทดแทนการซื้อเทคโนโลยีราคาแพงจากต่างประเทศ ทำให้เป็นประโยชน์ต่องานวิจัยและการพัฒนางานวิจัยของสถาบันฯ ต่อไป

Abstract

In the future research in various fields of KMITL such as a proof concept of Mathematics, Statistics, Biomedical analysis, fluid dynamic computation and other, need more computing power than it does nowadays. Super computer or high performance computer is indispensable for the researchers to solve these complicated problems. But supercomputer is very expensive and it has high cost to maintenance. Therefore, the concept of parallel computing, well known as computer clustering, has been implemented for these requirements. The computer clustering performs high performance of computing power and it is the low price solution. It can help us to reduce cost of investment in new technology.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ *1*

โครงสร้างของคอมพิวเตอร์ขนานแบบคลัสเตอร์

1.1 บทนำ

ในบทนี้จะเป็นการกล่าวถึงรายละเอียดกว้าง ๆ ของระบบการประมวลผลแบบขนาน ที่เรียกว่า คอมพิวเตอร์ขนานแบบคลัสเตอร์ โดยอาจจะไม่กล่าวถึงรายละเอียดเชิงวิชาการมากนัก เพราะต้องการให้เห็นภาพองค์ประกอบโดยรวมเกี่ยวกับการทำงานของระบบนี้มากกว่าการอ้างอิงเชิงทฤษฎี ซึ่งเนื้อหาภายในบทจะเริ่มจากการชี้ให้เห็นความสำคัญของเครื่องคอมพิวเตอร์ประสิทธิภาพสูงในการคำนวณทางวิทยาศาสตร์ ชนิดและโครงสร้างของคอมพิวเตอร์แบบขนาน การจำแนกชนิดของคอมพิวเตอร์แบบคลัสเตอร์ และชี้ให้เห็นความแตกต่างระหว่าง คอมพิวเตอร์แบบคลัสเตอร์ กริด และเครือข่ายแบบ P2P ส่วนประกอบโดยรวมของระบบ โครงสร้างของระบบคลัสเตอร์ เครื่องมือและโปรแกรมอัตถะประโยชน์ หัวข้อสุดท้ายจะเป็นการอธิบายสรุปเกี่ยวกับการจัดการระบบคลัสเตอร์

1.2 ความสำคัญของคอมพิวเตอร์ความเร็วสูง

ในการคำนวณทางวิทยาศาสตร์ วิศวกรรมศาสตร์ หรือทางด้านอื่น ๆ ในปัจจุบันและอนาคตจำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพที่สูงมากขึ้นในการประมวลผล ไม่ว่าจะเป็นการสร้างภาพเชิงปริมาตรทางการแพทย์ขนาดใหญ่มาก ดังเช่นในงานวิจัยนี้ หรืองานทางด้านอื่น ๆ เช่น การจำลองสภาวะอากาศของโลก การวิจัยตัวยาใหม่ ๆ การวิเคราะห์รูปแบบรหัสพันธุกรรมหรือ DNA ของเชื้อโรคที่ไม่เคยรู้จัก อย่างเช่น การวิจัยรหัสพันธุกรรมของไวรัสต้นเหตุของโรค Severe Acute Respiratory Syndrome หรือ SARS ที่เพิ่งเกิดขึ้นในเดือนมีนาคม พ.ศ. 2546 ที่ผ่านมา การประมวลผลคลื่นสัญญาณต่าง ๆ ที่ได้รับจากอวกาศในชื่อโครงการ SETI@home (The Search for Extraterrestrial Intelligence :SETI: http://setiathome.berkeley.edu) หรือโครงการวิจัย TeraGrid (http://www.teragrid.org) ซึ่งเป็นโครงการคอมพิวเตอร์แบบกริด เพื่อใช้ประโยชน์ร่วมกันในการประมวลผลงานทางด้านวิทยาศาสตร์และเทคโนโลยี และสุดท้ายโครงการงานวิจัยเกี่ยวกับสภาวะแผ่นดินไหวของโลก (http://www.neesgrid.org) เป็นต้น

ดังนั้นจะเห็นได้ว่าถ้าหากเรามีเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพในการคำนวณสูง ผลลัพธ์ที่ต้องการของปัญหาจะถูกคำนวณออกมาได้รวดเร็ว ทำให้เราสามารถเข้าใจปัญหา และสามารถนำผลไปวิเคราะห์เพื่อใช้งานต่อไปได้ถูกต้องและง่ายยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และหนึ่งในบรรดาคอมพิวเตอร์ความเร็วสูงที่นิยมนำมาใช้ในงานวิจัยต่าง ๆ คือ ระบบคอมพิวเตอร์แบบคลัสเตอร์ ซึ่งเป็นเทคโนโลยีที่ได้รับการพัฒนาให้ดีขึ้นโดยลำดับ และเริ่มถูกนำมาใช้ในงานวิจัยอย่างกว้างขวาง แต่อาจจะมีคำถามว่าทำไมถึงต้องมีการสร้างระบบคลัสเตอร์ขึ้นมาในโลกปัจจุบันซึ่งมีซุปเปอร์คอมพิวเตอร์ หรือคอมพิวเตอร์ความเร็วสูงที่ผลิตในเชิงการค้าได้อย่างมีประสิทธิภาพอย่างมากมายแล้ว คำตอบสั้น ๆ ก็คือ เรื่องของงบประมาณ ซุปเปอร์คอมพิวเตอร์นั้นมีราคาที่ค่อนข้างแพงมากเมื่อเปรียบเทียบกับคอมพิวเตอร์แบบอื่น ๆ หรือเปรียบเทียบราคากันไม่ได้เลยกับคอมพิวเตอร์ส่วนบุคคลทั่วไป ดังนั้นการสร้างระบบคลัสเตอร์จึงน่าจะเป็นอีกทางเลือกหนึ่งที่เหมาะสมสำหรับห้องวิจัยและมหาวิทยาลัยซึ่งมีงบประมาณไม่มากนัก ในการที่จะมีคอมพิวเตอร์ความเร็วสูงแต่ไม่ต้องลงทุนในด้านงบประมาณมากนักเพื่อใช้ในการคำนวณ เพราะระบบคลัสเตอร์จะเป็นการนำเอาเทคโนโลยีที่ราคาถูกกว่ามาทำงานร่วมกันให้เกิดประสิทธิภาพสูงสุด และเพียงพอสำหรับงานวิจัยต่าง ๆ

อุปกรณ์ที่นำมาใช้ต่อร่วมกันของคอมพิวเตอร์แบบคลัสเตอร์สามารถจะเป็นอุปกรณ์ที่เก่าหรือใหม่ก็ได้ ขอเพียงแค่เป็นอุปกรณ์ที่ยังมีสภาพดีอยู่และสามารถทำงานร่วมกับอุปกรณ์อื่น ๆ ได้ ตัวอย่าง ในสมัยแรกที่มีการพัฒนาระบบแบบคลัสเตอร์ขึ้นมานั้น เครื่องคอมพิวเตอร์ในระบบก็ใช้หน่วยประมวลผลของบริษัทอินเทล รุ่น Intel DX4 ในการทำงาน ฟังดูอาจจะไม่หรูหรามากนัก แต่ระบบนี้ก็สามารถทำงานได้จริง และทำให้งานวิจัยต่าง ๆ สำเร็จลงได้ และเป็นต้นแบบให้กับคอมพิวเตอร์แบบคลัสเตอร์ในปัจจุบันที่มีเทคโนโลยีความเร็วของหน่วยประมวลผลเพิ่มมากขึ้น ดังนั้นจึงเห็นได้ว่าในการสร้างระบบคลัสเตอร์ไม่จำเป็นต้องทำการลงทุนกับฮาร์ดแวร์ใหม่ สามารถใช้อุปกรณ์ที่มีอยู่แล้วได้เลย หรือถ้าหากมีงบประมาณเพียงพอก็สามารถที่จะเพิ่มหน่วยประมวลผลที่มีความเร็วสูงมากขึ้นเข้ามาในระบบเพิ่มเติมได้ด้วย

1.3 ชนิดและโครงสร้างของคอมพิวเตอร์แบบขนาน

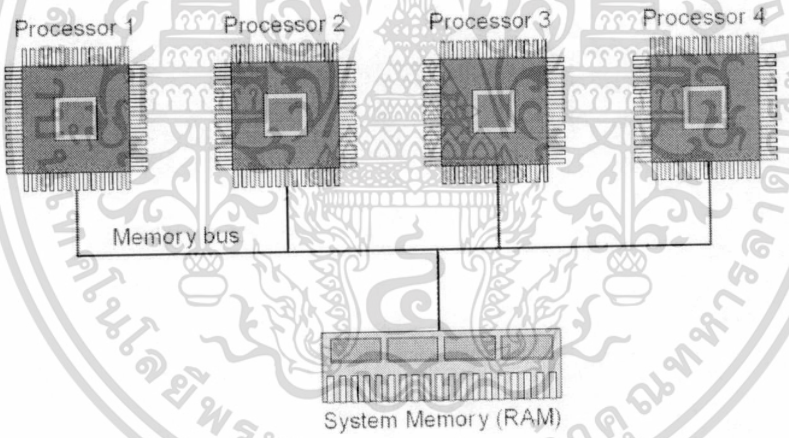
เครื่องคอมพิวเตอร์ประสิทธิภาพสูงที่กล่าวในหัวข้อที่ผ่านมานั้น ทั้งซุปเปอร์คอมพิวเตอร์และคอมพิวเตอร์แบบคลัสเตอร์ ล้วนแล้วแต่เป็นเครื่องคอมพิวเตอร์คำนวณแบบขนานทั้งสิ้น เพราะเทคโนโลยีนี้เป็นการใช้ประโยชน์จากการนำหลาย ๆ หน่วยประมวลผลมาทำงานร่วมกันให้เกิดประสิทธิภาพในการคำนวณมากกว่าการใช้เพียงแค่หน่วยประมวลผลเดียว ซึ่งหลักการการประมวลผลแบบขนานเป็นวิธีการในการแบ่งปัญหาขนาดใหญ่ออกเป็นปัญหาย่อยๆ แล้วทำการแก้ปัญหาเหล่านั้นไปพร้อม ๆ กัน ในปัจจุบันระบบประมวลผลแบบขนานถูกใช้อย่างแพร่หลายทั้งในการคำนวณทางวิทยาศาสตร์และการประยุกต์ใช้กับงานทั่วๆไป จากความต้องการความสามารถในการประมวลผลที่มากขึ้นนี้ทำให้มีความต้องการระบบคอมพิวเตอร์ที่มีความเร็วสูงขึ้น ราคาถูกลง และมีการพัฒนาอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับหลักการทำงานของเครื่องคอมพิวเตอร์แบบขนานนั้นกล่าวคือ การที่เราสามารถนำเอาหลายๆ หน่วยประมวลผลมาทำงานร่วมกันหรือคำนวณปัญหาเดียวกัน โดยที่หน่วยประมวลผลเหล่านี้ต้องสามารถทำการสื่อสารระหว่างกันได้ ไม่ว่าจะโดยวิธีการใช้หน่วยความจำร่วมกัน (Share Memory) หรือโดยวิธีการส่งข้อความระหว่างหน่วยประมวลผลต่อหน่วยประมวลผล (Message Passing) [Lester,1993] ซึ่งเครื่องคอมพิวเตอร์ที่ใช้หลักการหน่วยความจำร่วมนี้จะเรียกว่า ระบบแบบหลายหน่วยประมวลผล (Multiprocessor) และระบบคอมพิวเตอร์ที่ใช้หลักการส่งผ่านข้อความจะถูกเรียกว่า ระบบแบบหลายเครื่องประมวลผล (Multi computer) ซึ่งแต่ละหน่วยประมวลผลจะมีหน่วยความจำเป็นของตัวเองทำให้สามารถเรียกโครงสร้างแบบนี้ได้อีกอย่างหนึ่งว่า ระบบคอมพิวเตอร์แบบกระจาย (Distributed Computing) ซึ่งรายละเอียดของคอมพิวเตอร์ขนานแต่ละแบบสามารถกล่าวได้โดยละเอียดดังนี้

1.3.1 ระบบแบบหลายหน่วยประมวลผล (Multiprocessor)

หลักการทำงานของเครื่องคอมพิวเตอร์แบบหลายหน่วยประมวลผลสามารถแสดงรูปแบบการเชื่อมต่อได้ดังรูปต่อไปนี้



รูปที่ 1.1 โครงสร้างการทำงานของสถาปัตยกรรมแบบหลายหน่วยประมวลผล

จากรูปที่ 5.1 จะเห็นได้ว่าแต่ละหน่วยประมวลผลสามารถทำงานไปพร้อม ๆ กันได้ เพราะทุกหน่วยประมวลผลสามารถที่จะเข้าถึงหน่วยความจำส่วนกลางผ่านทางช่องสัญญาณบัส (Bus) ได้อย่างเท่าเทียมกัน การทำงานของหน่วยประมวลผลในระบบแบบนี้จะทำงานคล้าย ๆ กับระบบที่มีหน่วยประมวลผลเดียว คือ จะทำการอ่านค่าข้อมูลมาจากหน่วยความจำส่วนกลาง ทำการคำนวณค่าใหม่ แล้วใส่กลับเข้าไปที่หน่วยความจำส่วนกลางเช่นเดิม แต่ว่าทุกหน่วยประมวลผลจะสามารถทำงานต่าง ๆ เหล่านี้ได้อย่างพร้อม ๆ กัน ฉะนั้นจะได้ว่า หากในระบบมีหน่วยประมวลผล N หน่วย ระบบนี้จะมีความเร็วสูงที่สุดที่เป็นไปได้ เท่ากับ N เท่าของเครื่องที่มีหน่วยประมวลผลเพียงตัวเดียว แต่ในทาง

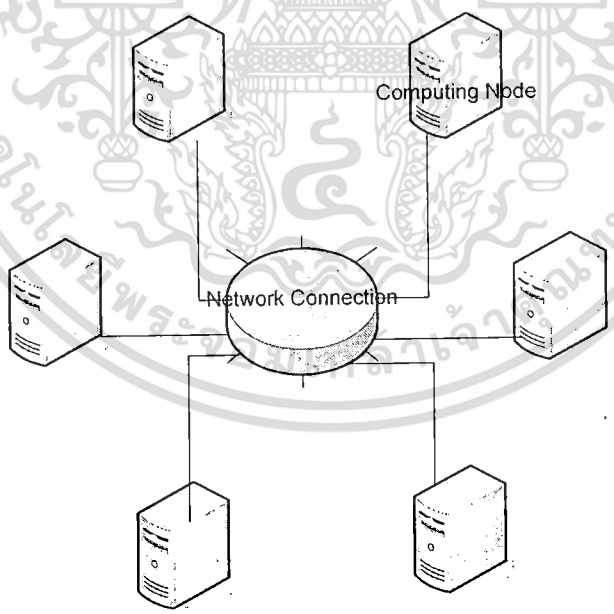
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิบัติความเร็วที่เพิ่มขึ้นต่อหน่วยประมวลผลมักจะไม่ได้เป็นไปตามทฤษฎีมากนัก เพราะปัญหาหลาย ๆ อย่างในทางปฏิบัติ

ปัญหาของระบบนี้ที่พบได้บ่อยคือ ความคับคั่งในการเข้าถึงหน่วยความจำ (Memory Contention) ซึ่งปัญหาเกิดขึ้นเนื่องมาจากในระบบมีหลายหน่วยประมวลผลที่ต้องการเข้าถึงหน่วยความจำเพื่ออ่านหรือเขียน ทำให้หน่วยความจำไม่สามารถตอบสนองต่อความต้องการในการอ่านหรือเขียนทั้งหมดพร้อมกันได้ จึงมีผลทำให้เกิดปัญหานี้ขึ้น ดังนั้นการออกแบบโครงสร้างของระบบที่ดีจึงมีส่วนสำคัญที่จะช่วยแก้ปัญหาเรื่องความคับคั่งในการเข้าถึงหน่วยความจำได้ ทางออกหนึ่งก็คือการกำหนดให้แต่ละหน่วยประมวลผลมีหน่วยความจำชั่วคราว (Local cache memory) เป็นของตัวเอง เพื่อช่วยให้หน่วยประมวลผลสามารถเก็บข้อมูลในการคำนวณหรือผลลัพธ์หลังจากการคำนวณได้มากขึ้น .ก่อนที่จะติดต่อกับหน่วยความจำส่วนกลางต่อไป

1.3.2 ระบบแบบหลายเครื่องประมวลผล (Multi computer)

การหลีกเลี่ยงปัญหาความคับคั่งในการเข้าถึงหน่วยความจำในหัวข้อที่ผ่านมา สามารถแก้ไขได้อีกวิธีหนึ่งนั่นคือ การกระจายให้ทุก ๆ เครื่องมีหน่วยความจำขนาดใหญ่เป็นของตัวเอง แล้วทำการติดต่อสื่อสารกันผ่านทางเครือข่ายโดยใช้วิธีการส่งข้อความ ซึ่งระบบนี้มักจะถูกเรียกว่าระบบหลายเครื่องประมวลผลหรือ Multi computer โดยสามารถแสดงรูปแบบการเชื่อมต่อของระบบได้ดังรูปต่อไปนี้



รูปที่ 1.2 โครงสร้างการทำงานของสถาปัตยกรรมแบบหลายเครื่องประมวลผล

จากรูปที่ 5.2 แต่ละเครื่องจะสามารถทำงานได้โดยอิสระต่อกัน และเป็นเครื่องคนละรุ่นคนละรูปแบบได้โดยอิสระ (Heterogeneous) หรือจะเป็นเครื่องรุ่นเดียวกันแบบเดียวกันทั้งหมดก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Homogeneous) โดยเมื่อทำการคำนวณจะเก็บข้อมูลเกี่ยวกับการคำนวณไว้ที่หน่วยความจำของแต่ละเครื่องเอง แล้วทำการสื่อสารข้อมูลและรับส่งค่าผลลัพธ์กับเครื่องอื่น ๆ ผ่านทางเครือข่ายโดยวิธีการส่งผ่านข้อความ ซึ่งหลักการนี้จะแตกต่างกับหัวข้อที่ผ่านมาทั้งในด้านโครงสร้างของฮาร์ดแวร์และวิธีการเขียนโปรแกรม เพราะในหัวข้อที่ผ่านมาทุกหน่วยประมวลผลจะใช้หน่วยความจำที่เดียวกัน แต่ในโครงสร้างแบบนี้หน่วยประมวลผลแต่ละหน่วยจะมีพื้นที่เก็บข้อมูลเป็นของตัวเองแยกจากกัน หรืออาจจะกล่าวได้ว่ามีหน่วยความจำสองรูปแบบ คือ หน่วยความจำแบบส่วนตัวของหน่วยประมวลผลหรือหน่วยความจำแบบโลคอล (Local memory) และหน่วยความจำของหน่วยประมวลผลอื่นหรือหน่วยความจำแบบรีโมต (Remote memory) ซึ่งการเข้าถึงหน่วยความจำแบบโลคอลนั้นแต่ละหน่วยประมวลผลสามารถเข้าถึงได้โดยตรง แต่ถ้าหากเป็นการเข้าถึงเพื่ออ่านหรือส่งค่าให้หน่วยความจำอื่น ๆ ต้องกระทำผ่านเครือข่ายตามวิธีการส่งผ่านข้อความ

โครงสร้างการเชื่อมต่อของระบบแบบหลายเครื่องประมวลผลนี้มีอยู่ด้วยกันหลายรูปแบบ ซึ่งรูปแบบการเชื่อมต่อเครือข่ายแต่ละแบบนั้นมีวัตถุประสงค์เพื่อให้ได้ประสิทธิภาพการคำนวณให้สูงมากที่สุดเท่าที่จะเป็นไปได้ โดยสามารถแยกตามรูปแบบและโครงสร้างการเชื่อมต่อออกได้เป็นสามประเภทดังนี้

- **คอมพิวเตอร์ระบบคลัสเตอร์ (Clustering Computer)**

แนวความคิดของระบบคลัสเตอร์เกิดขึ้นในปี ค.ศ. 1960 โดยบริษัท IBM ต้องการเชื่อมต่อระบบเมนเฟรม (Mainframe) ขนาดใหญ่เข้าด้วยกันเพื่อใช้ในเชิงการค้า แต่ในตอนนั้นระบบคลัสเตอร์ยังไม่เป็นที่รู้จักอย่างแพร่หลายเนื่องจากข้อจำกัดทางด้านราคาและความแพร่หลายของฮาร์ดแวร์ (Hardware) นอกจากนั้นยังขาดเครื่องมือ (Tools) มาตรฐานสำหรับการเชื่อมต่อเครื่องในระบบเข้าด้วยกัน จนในช่วงกลางปี 1980 เป็นช่วงที่เทคโนโลยีระบบคลัสเตอร์สามารถเกิดขึ้นได้เนื่องจากสามารถสร้าง ไมโครโพรเซสเซอร์ที่มีประสิทธิภาพได้ และเกิดระบบเครือข่ายความเร็วสูงขึ้น ซึ่งเครื่องคอมพิวเตอร์ในระบบเครือข่ายสามารถสื่อสารกันได้ภายในเวลาไม่เกินหนึ่งมิลลิวินาที นอกจากนั้นความต้องการความเร็วในการประมวลผลงานทางวิทยาศาสตร์และวิศวกรรมศาสตร์ซึ่งมีมากขึ้นอย่างต่อเนื่องก็มีส่วนผลักดันให้เกิดเทคโนโลยีคลัสเตอร์ขึ้น

ระบบคลัสเตอร์ได้เกิดขึ้นอย่างเป็นรูปธรรมในปี 1994 เมื่อ Thomas Sterling และ Don Becker จาก CESDIS (The Center of Excellence in Space Data and Information Sciences) ต้องการที่จะวิเคราะห์ข้อมูลจากอวกาศที่มีจำนวนมากและซับซ้อนซึ่งต้องใช้เครื่องคอมพิวเตอร์ที่สามารถประมวลผลได้เร็วมากๆและในขณะนั้นมีเพียงเครื่องซุปเปอร์คอมพิวเตอร์ซึ่งมีราคาแพงแต่เนื่องจากงบประมาณที่มีอยู่จำกัดทำให้ Sterling และ Becker มีแนวความคิดที่จะใช้ระบบคลัสเตอร์มาทำการประมวล โดยได้สร้างระบบคลัสเตอร์จากเครื่องคอมพิวเตอร์ซึ่งใช้ชิพ Intel DX4 เป็นหน่วยประมวลผลกลาง (CPU) และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบอื่นๆซึ่งสามารถหาได้ง่ายจากท้องตลาดและได้ตั้งชื่อระบบคลัสเตอร์นี้ว่า Beowulf (Sterling and Becker, 1994)

ระบบคลัสเตอร์มักจะเป็นระบบที่แต่ละหน่วยประมวลผลหรือโหนดการคำนวณ (Computing node) อยู่ใกล้กัน หรืออยู่ในพื้นที่ใกล้เคียงกัน แล้วทำการเชื่อมต่อระหว่างโหนดต่อโหนดหรือแต่ละหน่วยประมวลผลผ่านทางเครือข่าย เช่น Gigabit Ethernet, Myrinet, InfiniBand, Quadrics หรือ เครือข่ายรูปแบบอื่น ๆ ซึ่งต้องเป็นเทคโนโลยีเครือข่ายที่มีความเร็วสูง และค่า Latency Time ต่ำ เพื่อให้การติดต่อสื่อสารระหว่างกันสามารถเป็นไปได้อย่างรวดเร็ว และสูญเสียเวลาในเครือข่ายน้อยที่สุดเสมือนว่าทุกหน่วยประมวลผลอยู่ใกล้กันมากจนเกือบจะเหมือนอยู่บนแผงวงจรรวม Mother Board เดียวกัน ทำให้การส่งผ่านข้อความเพื่อแลกเปลี่ยนข้อมูลต่าง ๆ ระหว่างคำนวณเป็นไปอย่างรวดเร็ว

สำหรับรูปแบบการเชื่อมต่อและโครงสร้างการทำงานต่าง ๆ ทั้งในด้านซอฟแวร์และฮาร์ดแวร์จะกล่าวโดยละเอียดในหัวข้อต่อ ๆ ไปของบทนี้

- **กริดคอมพิวเตอร์**

เนื่องจากมีความต้องการคอมพิวเตอร์ความเร็วสูงเพื่อใช้ในการคำนวณทางคณิตศาสตร์ที่มีความยุ่งยากซับซ้อนมาก ๆ ตัวอย่างเช่น การคำนวณทางไฟไนต์อีลิเมนต์ (Finite Element) การวิเคราะห์โครงสร้างรหัสพันธุกรรมของไวรัส การสังเคราะห์โปรตีน ซึ่งเป็นงานที่จำเป็นต้องใช้ประสิทธิภาพของคอมพิวเตอร์ในการคำนวณสูงมาก แต่ลักษณะงานเหล่านี้ไม่ต้องการความเร็วในการสื่อสารระหว่างหน่วยประมวลผลมากนัก หน่วยประมวลผลสามารถสื่อสารระหว่างกันด้วยความเร็วไม่สูงมาก การแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผลมีเพียงเล็กน้อยเท่านั้น ซึ่งจากความต้องการและเงื่อนไขเหล่านี้ทำให้มีความพยายามในการพัฒนาระบบคอมพิวเตอร์ขนาดใหญ่ขึ้น โดยเป็นการรวมเอาระบบคลัสเตอร์ หลาย ๆ คลัสเตอร์เข้าไว้ด้วยกันทำให้สามารถทำการคำนวณร่วมกันได้

กริดคอมพิวเตอร์เป็นระบบคอมพิวเตอร์แบบกระจายที่มีขนาดใหญ่มาก และมีความเร็วในการคำนวณสูงมากเช่นกัน ทำให้การคำนวณปัญหาขนาดใหญ่สามารถเสร็จได้ในเวลาอันรวดเร็ว ตัวอย่างของกริดคอมพิวเตอร์ขนาดใหญ่เช่น โครงการ TeraGrid (http://www.teragrid.org) ซึ่งเป็นการรวมเอาระบบคลัสเตอร์จากหลาย ๆ ที่ให้มาทำงานร่วมกัน ทำให้ความสามารถในการประมวลผลของระบบนี้มีความเร็วสูงขึ้น

การเชื่อมต่อในระบบกริดคอมพิวเตอร์นั้นจะทำการเชื่อมคลัสเตอร์แต่ละที่เข้าด้วยกันโดยใช้เครือข่ายที่มีอยู่แล้วในปัจจุบัน นั่นคือเครือข่ายอินเทอร์เน็ตนั่นเอง ทำให้แต่ละไซต์ (Site) สามารถอยู่กระจายในที่ต่าง ๆ ได้ทั่วโลก ขอเพียงแค่เชื่อมต่อกับระบบอินเทอร์เน็ต ก็จะสามารถทำงานร่วมกันในระบบของกริดได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดการทรัพยากรต่าง ๆ ในระบบกริดคอมพิวเตอร์จะต้องมีซอฟแวร์ (Software) ที่เรียกว่า Grid Middleware เป็นตัวกลางในการจัดการและเชื่อมการทำงานเข้าด้วยกัน ซอฟท์แวร์ที่ได้รับความนิยมในปัจจุบันตัวอย่างเช่น Globus Project (http://www.globus.org)

- **เครือข่ายแบบ P2P (Peer-to-Peer)**

โครงข่ายคอมพิวเตอร์แบบ Peer-to-Peer เป็นคอมพิวเตอร์ความเร็วสูงอีกรูปแบบหนึ่งที่ได้รับความนิยมมากขึ้นเรื่อย ๆ เพราะเครื่องที่ทำงานในระบบคือเครื่องของผู้ใช้งานทั่ว ๆ ไป ที่เชื่อมต่ออยู่กับระบบอินเทอร์เน็ต เพราะฉะนั้นขนาดและจำนวนของหน่วยประมวลผลในระบบจึงสามารถเพิ่มขยายได้เรื่อย ๆ ตามจำนวนของผู้ที่เข้าร่วมในโครงการ หรืออาจกล่าวได้ว่าคอมพิวเตอร์ทุกเครื่องที่ต่ออยู่กับระบบอินเตอร์เน็ตสามารถที่จะร่วมประมวลผลในรูปแบบของเครือข่ายประเภทนี้ได้ งานวิจัยในปัจจุบันที่ใช้การประมวลผลโดยใช้เครือข่ายแบบ P2P มีตัวอย่างเช่น โครงการการค้นหาสิ่งมีชีวิตจากอวกาศหรือ SETI@Home ซึ่งเคยได้กล่าวไปแล้ว โดยโครงการนี้จะทำการรับเอาคลื่นวิทยุจากจานรับสัญญาณจากอวกาศมาทำการประมวลผลเพื่อหาสัญญาณที่ไม่ใช่รูปแบบของสัญญาณในธรรมชาติทั่วไป
อีกโครงการหนึ่งเช่น การวิจัยตัวยาชนิดใหม่โดยใช้โครงสร้างแบบกริดและเครือข่ายแบบ P2P ในชื่อโครงการ DesignDrug@Home (http://www.gridbus.org/vlab/)

คอมพิวเตอร์ความเร็วสูงแบบหลายหน่วยประมวลผลทั้งสามชนิด สามารถที่จะแสดงให้เห็นถึงความแตกต่างของแต่ละรูปแบบได้ดังตารางต่อไปนี้

ตารางที่ 1.1 เปรียบเทียบความแตกต่างของคอมพิวเตอร์แบบกระจาย (Buyya, 1999)

Characteristic	Cluster	Grid	P2P
Population	Commodity Computers	High-end computers	Edge of network (desktop PC)
Ownership	Single	Multiple	Multiple
Discovery	Membership Services	Centralised Index & Decentralised Info	Decentralized
User Management	Centralised	Decentralised	Decentralised
Resource management	Centralized	Distributed	Distributed
Allocation/Scheduling	Centralised	Decentralised	Decentralised
Inter-Operability	VIA based?	No standards yet	No standards
Single System Image	Yes	No	No
Scalability	100s	1000?	Millions? [@Home]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Capacity	Guaranteed	Varies, but high	Varies
Throughput	Medium	High	Very High
Speed(Latency Time. Bandwidth)	Low, high	High, Low	High, Low

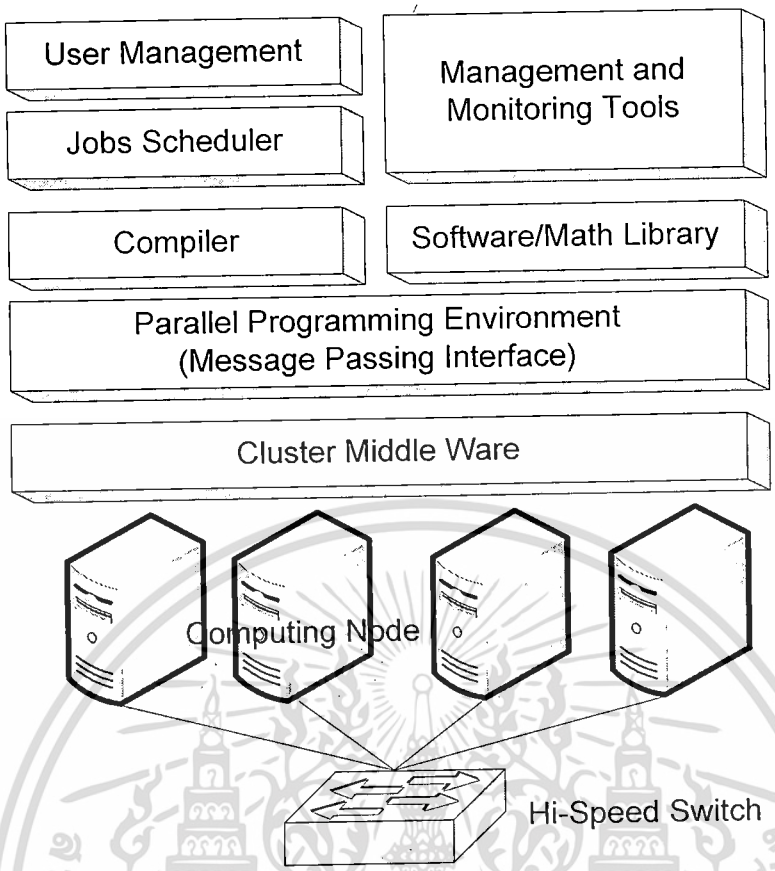
ซึ่งจากที่กล่าวมาทั้งหมดจะเห็นได้ว่าคอมพิวเตอร์ความเร็วสูงแบบหลายเครื่องประมวลผลเหล่านี้เริ่มเข้ามามีบทบาทในงานวิจัยต่าง ๆ ในยุคปัจจุบันมากยิ่งขึ้น จากเดิมที่งานวิจัยหลาย ๆ อย่าง ต้องทำการทดลองในห้องปฏิบัติการ หรือแสดงได้เพียงแค่รูปแบบสมการเพียงอย่างเดียว แต่ในปัจจุบันสามารถที่จะจำลองการทำงานหรือหาผลลัพธ์ได้บนคอมพิวเตอร์ความเร็วสูงเหล่านี้ ทำให้งานทางด้านวิทยาศาสตร์และเทคโนโลยีหลาย ๆ ด้านมีการพัฒนาอย่างรวดเร็วมากขึ้น

1.4 ส่วนประกอบโดยรวมของคอมพิวเตอร์แบบคลัสเตอร์

เราอาจกล่าวได้โดยง่ายว่าระบบคลัสเตอร์คือ กลุ่มของคอมพิวเตอร์ที่มีการเชื่อมต่อผ่านเครือข่ายความเร็วสูงและสามารถคำนวณงานที่ถูกแบ่งออกเป็นงานย่อย ๆ แล้วร่วมกันได้ ซึ่งรูปแบบของเครือข่ายและการเชื่อมต่อนี้มีด้วยกันหลายวิธีและหลายชนิดเครือข่าย ส่วนการที่จะทำให้กลุ่มของคอมพิวเตอร์สามารถทำงานร่วมกันได้นั้นจะต้องมีซอฟแวร์เป็นตัวกลางในการเชื่อมการทำงานของแต่ละหน่วยประมวลผลเข้าด้วยกัน (Spector, 2000) ซึ่งมาตรฐานที่นิยมใช้คือ ระบบการส่งผ่านข้อความหรือ Message Passing Interface และคลัสเตอร์มิดเดิลแวร์ (Cluster Middle Ware)

ส่วนประกอบโดยรวมทั้งหมดของระบบคอมพิวเตอร์แบบคลัสเตอร์นั้นจะประกอบไปด้วยส่วนต่าง ๆ ดังบล็อกไดอะแกรมด้านล่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 บล็อกไดอะแกรมแสดงส่วนประกอบของระบบคลัสเตอร์

จากรูปสามารถที่จะแบ่งการทำงานออกได้ทั้งหมดสามส่วนด้วยกันคือ

1. โครงสร้างพื้นฐานของระบบคลัสเตอร์ ซึ่งจะประกอบไปด้วยส่วนประกอบทางฮาร์ดแวร์ การเชื่อมต่อเครือข่าย ระบบปฏิบัติการ และคลัสเตอร์มิดเดิลแวร์
2. เครื่องมือและโปรแกรมอัตถะประโยชน์ ประกอบด้วย โปรแกรมสื่อสารระหว่างโหนดโดยการส่งข้อความ คอมไพเลอร์ และชุดคำสั่งทางคณิตศาสตร์
3. การจัดการระบบคลัสเตอร์ ประกอบด้วยตัวจัดการคิวงาน การจัดการสิทธิของผู้ใช้งานในระบบ และระบบตรวจสอบและจัดการคลัสเตอร์

ซึ่งจะได้กล่าวโดยละเอียดในหัวข้อที่ 5.6, 5.7 และ 5.8 ตามลำดับต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 การจำแนกคอมพิวเตอร์แบบคลัสเตอร์

เราสามารถที่จะจำแนกประเภทของระบบคลัสเตอร์ออกตามเกณฑ์ต่าง ๆ ได้ดังนี

1.5.1 จำแนกตามงานที่ประยุกต์ใช้

ระบบคลัสเตอร์สามารถนำเอาประยุกต์ใช้ได้กับงานที่หลากหลาย ไม่ว่าจะเป็นงานทางด้านการคำนวณหรืองานทางด้านการเป็นเครื่องแม่ข่ายให้บริการงานต่าง ๆ โดยสามารถแบ่งย่อยได้เป็นสองประเภทดังนี้

- ระบบคลัสเตอร์แบบประสิทธิภาพสูง (High Performance (HP) Clusters) มักจะถูกนำไปประยุกต์ใช้ในการคำนวณทางด้านวิทยาศาสตร์และคณิตศาสตร์ ระบบคลัสเตอร์แบบนี้จะถูกสร้างขึ้นมาให้มีความรวดเร็วในการคำนวณมากที่สุด ประสิทธิภาพของหน่วยประมวลผลจะต้องสูงเพียงพอ อีกทั้งเครือข่ายที่ใช้ในการเชื่อมต่อต้องมีคุณภาพดีมาก ประสิทธิภาพในการคำนวณจึงจะสูงตามไปด้วย ระบบคลัสเตอร์ที่ถูกสร้างขึ้นมาเพื่อใช้ในงานวิจัยนี้ก็เป็นคลัสเตอร์แบบความเร็วสูงเพื่อทำให้เวลาที่ใช้ในการสร้างภาพเชิงปริมาตรใช้เวลาในการสร้างภาพน้อยที่สุด
- ระบบคลัสเตอร์แบบเสถียรภาพสูง (High Availability (HA) Clusters) ระบบคลัสเตอร์แบบนี้จะเน้นไปทางด้านเครื่องแม่ข่ายที่ให้บริการงานต่าง ๆ เช่น ให้บริการเป็นเว็บเซอร์ฟเวอร์ (Web Server) หรือให้บริการพื้นที่เก็บข้อมูลบนเครือข่าย (Storage Server) เพื่อจะทำให้มั่นใจได้ว่าผู้ใช้งานทั่ว ๆ ไปจะสามารถเข้าถึงทรัพยากรต่าง ๆ ได้ตลอดเวลา การให้บริการไม่ได้ขึ้นอยู่เครื่องเซอร์ฟเวอร์เครื่องใดเครื่องหนึ่งเพียงเครื่องเดียว

  การสร้างระบบคลัสเตอร์แต่ละแบบนั้นจะมีข้อแตกต่างกัน วิธีการต่างกัน และซอฟต์แวร์ที่ใช้มีลักษณะแตกต่างกัน เพราะระบบหนึ่งต้องการความรวดเร็วในการคำนวณเพียงอย่างเดียว ส่วนอีกระบบหนึ่งต้องการความเชื่อถือในการเข้าถึงบริการได้ตลอดเวลา

1.5.2 จำแนกตามลักษณะของเครื่องในระบบ

ระบบคลัสเตอร์เกิดจากการเชื่อมต่อเครื่องคอมพิวเตอร์หลาย ๆ เครื่องเข้าด้วยกัน โดยเครื่องคอมพิวเตอร์เหล่านี้อาจจะมีลักษณะเหมือนกันทั้งหมดหรือไม่เหมือนกันเลยก็ได้ ซึ่งถ้าหากจำแนกระบบคลัสเตอร์ตามลักษณะของฮาร์ดแวร์ในระบบแล้ว สามารถที่จะจำแนกออกได้เป็นสองประเภทเช่นกันคือ

- ระบบคลัสเตอร์แบบเนื้อเดียว (Homogeneous Cluster) ระบบคลัสเตอร์แบบนี้เป็นระบบที่มีองค์ประกอบทางด้านฮาร์ดแวร์และซอร์ฟแวร์ที่ใช้ในแต่ละเครื่องเหมือนกันทั้งหมดได้แก่ หน่วยประมวลผลกลาง ชนิดและขนาดของหน่วยความจำ ชนิดและขนาดของฮาร์ดดิสก์ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของระบบปฏิบัติการ เป็นต้น โดยระบบคลัสเตอร์นี้เป็นแบบที่นิยมสร้าง เนื่องจากการบริหารจัดการระบบสามารถทำได้อย่างสะดวก นอกจากนั้นการเขียนโปรแกรมเพื่อทำการประมวลผลบนระบบคลัสเตอร์สามารถเขียนในครั้งเดียวแล้วทำงานได้กับทุก ๆ เครื่องในระบบ

- ระบบคลัสเตอร์แบบเนื้อผสม (Heterogeneous Cluster) ระบบคลัสเตอร์แบบนี้เป็นระบบที่มีความยืดหยุ่นสูงโดยสามารถสร้างจากเครื่องแบบใดก็ได้ที่สนับสนุนการประมวลผลแบบขนาน แต่ปัญหาของระบบนี้คือการสร้างโปรแกรมสำหรับประมวลผลแบบขนานจะมีความยุ่งยากและซับซ้อนมากขึ้นเนื่องจากต้องทำการสร้างโปรแกรมที่สามารถประมวลผลเฉพาะของแต่ละเครื่อง แต่ละระบบปฏิบัติการ เช่น เครื่องที่ใช้ระบบปฏิบัติการลีนุกส์ (Linux) และระบบที่ใช้ไมโครซอฟส์วินโดว์ (Microsoft Windows) เป็นระบบปฏิบัติการ โปรแกรมที่สร้างขึ้นมาอาจไม่สามารถใช้ด้วยกันได้ วิธีแก้ปัญหาอย่างหนึ่งคือสร้างโปรแกรมให้สนับสนุนมาตรฐานเช่น ANSI C เป็นต้น หรืออาจสร้างโปรแกรมที่สามารถประมวลผลได้บนทุกระบบโดยที่ใช้รหัสต้นฉบับ (Source Code) ตัวเดียวกัน เช่น ภาษาจาวา (Java)

  สิ่งหนึ่งที่ต้องให้ความสำคัญมากเช่นกันในระบบคลัสเตอร์แบบเนื้อผสมนี้ คือ การสมดุลงานในระบบ หรือ Load Balancing เพราะเนื่องจากว่าคอมพิวเตอร์ที่เชื่อมต่ออยู่กับระบบแบบนี้สามารถที่จะมีองค์ประกอบภายในแตกต่างกันได้ ทำให้คอมพิวเตอร์แต่ละโหนดมีความสามารถและประสิทธิภาพในการคำนวณผลได้รวดเร็วแตกต่างกัน เมื่อทำงานร่วมกันนั้นจะต้องการการกระจายงานที่เหมาะสม เพื่อให้เครื่องที่มีประสิทธิภาพมากกว่ามีโอกาสได้รับงานไปคำนวณมากกว่า ระบบทั้งหมดไม่ต้องรอค่าผลลัพธ์จากเครื่องที่มีความเร็วต่ำ ถ้าหากทำการสมดุลงานได้ดีแล้ว ประสิทธิภาพของระบบก็จะดีตามไปด้วย แต่ถ้าหากไม่คำนึงถึงเรื่องนี้แล้วประสิทธิภาพโดยรวมจะต่ำตามไปด้วย

1.6 โครงสร้างพื้นฐานของระบบคลัสเตอร์

ในการสร้างระบบคอมพิวเตอร์แบบคลัสเตอร์ จะต้องมีส่วนประกอบพื้นฐานของระบบซึ่งจะประกอบไปด้วยส่วนประกอบทางฮาร์ดแวร์ ระบบปฏิบัติการ และซอฟต์แวร์ที่ทำงานในระดับของเคอร์เนล (Kernel) ของระบบปฏิบัติการซึ่งเรียกซอฟต์แวร์นี้ว่า คลัสเตอร์มิดเดิลแวร์ และสุดท้ายต้องเชื่อมแต่ละโหนดเข้าด้วยกันผ่านทางเครือข่าย

1.6.1 ส่วนประกอบทางฮาร์ดแวร์

องค์ประกอบทางด้านฮาร์ดแวร์ที่จำเป็นในการพิจารณาเพื่อสร้างเป็นระบบคลัสเตอร์นั้นมีส่วนประกอบต่าง ๆ ดังนี้

- แผงวงจรรวม Mother Board หรือ System Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยประมวลผลหรือ CPUs
- หน่วยความจำชั่วคราว (RAM)
- หน่วยความจำหลักหรือ Disk Storage
- การ์ดเน็ตเวิร์ค (Network adapter)
- ตัวถังเครื่อง (Cases)

ซึ่งแต่ละโหนดควรจะมีส่วนประกอบต่าง ๆ ดังนี้เพื่อให้การทำงานมีประสิทธิภาพมากที่สุด ส่วนคุณสมบัติต่าง ๆ ทางด้านฮาร์ดแวร์ สามารถพิจารณาให้เหมาะสมตามความต้องการและงบประมาณที่มีอยู่

1.6.2 ระบบปฏิบัติการ

ระบบปฏิบัติการจะเป็นส่วนสำคัญอย่างมากต่อการทำงานของคอมพิวเตอร์และระบบคลัสเตอร์ เพราะคอมพิวเตอร์ในระบบคลัสเตอร์ต้องสามารถทำงานเองได้โดยอิสระ ไม่ขึ้นกับเงื่อนไขของเครื่องอื่น ถึงแม้มีเครื่องใดเครื่องหนึ่งในระบบหยุดไป คลัสเตอร์ก็ยังสามารถทำงานได้ ดังนั้นระบบปฏิบัติการจึงเป็นความจำเป็นในส่วนนี้ที่จะทำให้คอมพิวเตอร์แต่ละเครื่องสามารถทำงานโดยอิสระต่อกันได้ ระบบปฏิบัติการที่สามารถใช้กับระบบคลัสเตอร์ได้มีหลากหลาย เช่น Linux, Solaris, FreeBSD, Tru64 UNIX, HP-UX ซึ่งการเลือกใช้ระบบปฏิบัติการจะเป็นระบบปฏิบัติการตระกูลยูนิกส์หรือ Microsoft Windows ทั้งนี้ขึ้นอยู่กับความสะดวก ความเชี่ยวชาญและอุปกรณ์ที่เลือกใช้ อีกอย่างหนึ่งคือจะขึ้นอยู่กับซอร์ฟแวร์และชุดคำสั่งที่เลือกใช้ด้วย เช่น ถ้าหากเลือกใช้ OpenMosix (http://openmosix.sf.net) ที่ทำงานได้บนลีนุกซ์เท่านั้นก็จำเป็นต้องเลือกใช้ลีนุกซ์เป็นระบบปฏิบัติการอีกทั้งฮาร์ดแวร์ที่สนับสนุนด้วย

แต่โดยส่วนใหญ่ซอฟต์แวร์และชุดคำสั่งที่ถูกพัฒนาขึ้นมาเพื่อใช้ในระบบคลัสเตอร์นี้จะทำงานเข้ากันได้กับระบบปฏิบัติการตระกูลยูนิกส์เกือบทุกตัวอยู่แล้ว

1.6.3 คลัสเตอร์มิดเดิลแวร์

คลัสเตอร์มิดเดิลแวร์คือซอฟต์แวร์ที่ทำงานในระดับเดียวกันกับแกนหรือเคอร์เนล (Kernel) ของระบบปฏิบัติการ มีหน้าที่ในการกระจายโพรเซส (Process) จากโหนดหนึ่งไปยังโหนดอื่น ๆ ที่อยู่ในระบบคลัสเตอร์เดียวกัน โดยส่วนใหญ่คลัสเตอร์มิดเดิลแวร์นี้จะเป็นซอฟต์แวร์ที่เขียนเพิ่มเติมเข้าไปในแกนของระบบปฏิบัติการ เพื่อให้ทุกเครื่องที่อยู่ในระบบคลัสเตอร์เสมือนเป็นเครื่องคอมพิวเตอร์หลายหน่วยประมวลผลขนาดใหญ่ ที่เสมือนมีหน่วยความจำ หน่วยประมวลผลอยู่ที่เดียวกันเหมือนกับระบบคอมพิวเตอร์แบบ SMP (Symmetric Multi Processor) หรือ MMP (Massive Multi Processor) โดยมีเครือข่ายเป็นช่องทางสำหรับการติดต่อคล้ายกับบัส (Bus) ภายใน โพรเซสที่เกิดในโหนดหนึ่งสามารถข้ามหรือกระจายไปยังโหนดอื่น ๆ ได้โดยสะดวก การเขียนโปรแกรมเพื่อใช้งานบนระบบที่ใช้คลัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เตอร์มิดเดิลแวร์นี้ก็สามารถทำได้ง่าย เพราะผู้ใช้ไม่ต้องศึกษาเพิ่มเติมเกี่ยวกับการใช้งานชุดคำสั่งเหมือนกับการเขียนโปรแกรมบนสภาวะการโปรแกรมแบบขนาน (Parallel Programming Environment) ที่จะได้กล่าวในหัวข้อถัดไป การสมดุลงานหรือ Load Balancing เอง ผู้ใช้งานก็ไม่จำเป็นต้องคำนึงถึง กล่าวคือทุกอย่างทั้งการกระจายงาน การสมดุลงาน คลัสเตอร์มิดเดิลแวร์จะเป็นตัวจัดการให้ทั้งหมด แต่ทั้งนี้ในเรื่องของประสิทธิภาพและ ความรวดเร็วแล้ว อาจจะด้อยกว่าการเขียนโปรแกรมบนสภาวะการโปรแกรมแบบขนานเพราะในสภาวะแวดล้อมแบบขนาน ผู้ใช้จะต้องทำการเขียนลำดับขั้นตอนทุกอย่างด้วยตัวเอง ทำให้ความเข้าใจในเนื้องานมีมากกว่า และทำให้โปรแกรมทำงานได้รวดเร็วกว่าด้วย ตัวอย่างของคลัสเตอร์มิดเดิลแวร์มีดังนี้

- BPROC (Beowulf distribute PROCess space) เป็นซอฟต์แวร์ที่ถูกพัฒนาโดยห้องวิจัย CESDIS ของ NASA ซึ่งเป็นต้นกำเนิดของระบบคลัสเตอร์แบบ Beowulf ระดับการทำงานของ BPROC นี้จะเป็นเพียงแค่โปรแกรมจำลองสภาวะของคลัสเตอร์เท่านั้น ไม่ได้ทำงานลึกลงไปในแกนของระบบปฏิบัติการอย่างแท้จริง แต่ก็ถือได้ว่าเป็นจุดเริ่มของคลัสเตอร์มิดเดิลแวร์รุ่นหลัง ๆ โดยหลักการทำงานก็ตรงไปตรงมาตามวิธีการของคลัสเตอร์มิดเดิลแวร์ คืออนุญาตให้มีการโอนย้าย Process จากเครื่องหนึ่งไปยังอีกเครื่องหนึ่ง และช่วยกันคำนวณจนได้ผลลัพธ์ที่ต้องการออกมาได้ เมื่อทำการคำนวณจะมีโหนดหนึ่งที่ถูกกำหนดให้เป็นโหนดหลักหรือ Master Node และส่วนที่เหลือทั้งหมดจะเรียกว่า สลาฟโหนด (Slave Node) โดยโหนดหลักหรือ Master Node จะทำการเก็บค่าหมายเลขโพรเซส (Process ID) ที่ทำงานอยู่ และรายละเอียดเกี่ยวกับโหนดอื่น ๆ ในระบบไว้ แล้วทำการกระจายโพรเซสไปยังโหนดอื่น ๆ ส่วน Slave Node เมื่อได้รับคำสั่งให้สร้างโพรเซสหรือได้รับงานมาจากมาสเตอร์ จะทำการคำนวณและส่งค่าผลลัพธ์กลับไปให้ยังมาสเตอร์โหนดต่อไป

- MOSIX เป็นซอฟต์แวร์ที่ถูกพัฒนาขึ้นมาโดย Hebrew University of Jerusalem เพื่อทำหน้าที่ทำให้เกิดการกระจายงานในระบบตามมาตรฐานของคลัสเตอร์มิดเดิลแวร์ ซึ่งแตกต่างจาก BPROC ตรงที่ MOSIX ไม่ได้เป็นเพียงแค่โพรเซสที่จำลองสภาวะเท่านั้น แต่ MOSIX ทำงานในระดับของแกนระบบปฏิบัติการ จึงทำให้มีประสิทธิภาพที่ดีกว่า สามารถควบคุมการกระจายงานต่าง ๆ ได้ดีกว่า คุณสมบัติที่สำคัญ ๆ คือ การกระจายโพรเซสในระบบคลัสเตอร์ การทำหน้าที่สมดุลงานในระบบให้อัตโนมัติ และจัดการหน่วยความจำของระบบทั้งหมดอย่างมีประสิทธิภาพ อันมีผลทำให้ระบบของ MOSIX สามารถขยายได้และมีประสิทธิภาพสูง การทำงานของ MOSIX จะเป็นไปโดยอัตโนมัติทุกอย่างโดยที่ผู้ใช้งานหรือผู้เขียนโปรแกรมไม่ต้องศึกษาเพิ่มเติมมากนัก ก็สามารถเขียนโปรแกรมให้ทำงานบนระบบ คลัสเตอร์ได้อย่างมีประสิทธิภาพ

79727

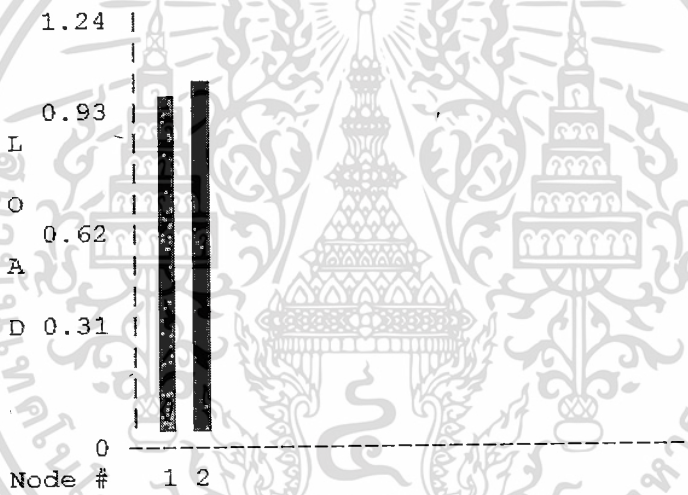
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบจะไม่ได้เป็นการควบคุมจากศูนย์กลาง (Decentralized) เหมือนกับ BPROC โดยจะไม่มีการแบ่งว่าโหนดไหนเป็น Master หรือ Slave Node แต่ทุกโหนดสามารถกระจายงานข้ามไปมาระหว่างกันได้อย่างสะดวกและง่ายดาย

ตัวอย่างโปรแกรม Shell Script ของ UNIX ที่ทำการทดสอบบนระบบคลัสเตอร์โดยใช้ OpenMOSIX (http://openmosix.sf.net) เป็นคลัสเตอร์มิดเดิลแวร์

```
1:  #!/bin/bash
    for x in 1 2 3 4
    do
      awk 'BEGIN {for(i=0;i<10000;i++)for(j=0;j<10000;j++);}' &
5:  done
```

และรูปที่ 5.4 คือ กราฟแสดงการกระจายงานระหว่างที่ทำการประมวลผล



รูปที่ 1.4 กราฟแสดงการกระจายงานระหว่างที่ประมวลผลในระบบคลัสเตอร์มิดเดิลแวร์

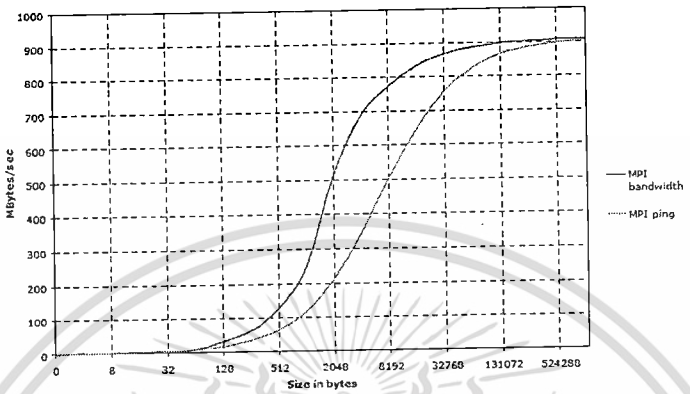
1.6.4 การเชื่อมต่อเครือข่าย

การสื่อสารระหว่างแต่ละเครื่องในระบบคลัสเตอร์จะทำผ่านระบบเครือข่ายความเร็วสูงซึ่งอุปกรณ์เครือข่ายแต่ละชนิดจะมีความเร็วและราคาแตกต่างกันไป ตัวอย่างอุปกรณ์เครือข่ายที่นิยมนำมาใช้ในการสร้างระบบคลัสเตอร์ มีดังนี้

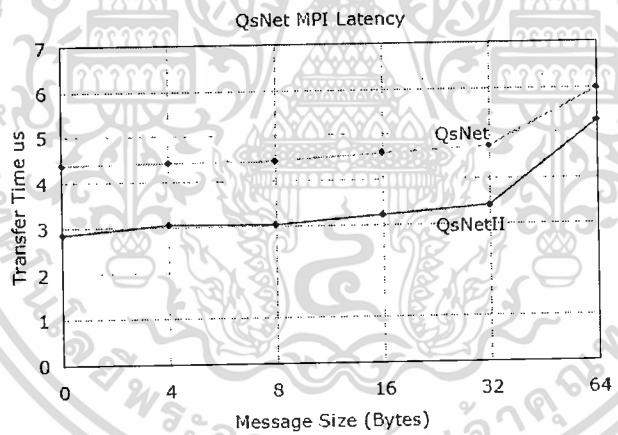
- Ethernet (http://www.10gea.org) ในปัจจุบันอุปกรณ์ Ethernet นั้นได้ถูกพัฒนาให้มีความเร็วในการรับส่งข้อมูลสูงมากขึ้นจนถึงระดับ Gigabit Ethernet หรือ 10 Gigabit Ethernet คือ มีความเร็วในการส่งผ่านข้อมูลประมาณ 1-10 พันล้านบิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Myrinet (http://www.myri.com) มีความเร็วในการส่งผ่านข้อมูลประมาณ 2 พันล้านบิตต่อวินาที (Gigabit per second) และมีค่า Latency Time ต่ำกว่าเครือข่ายแบบ Ethernet มาก แต่มีราคาแพงกว่า Fast Ethernet
- Quadrics (http://www.quadrics.com) มีความเร็วในการส่งข้อมูลอยู่ที่ 340-900 MB/second หรือประมาณ 2.65-7 Gbit/sec และค่า Latency Time มีค่าต่ำมาก



รูปที่ 1.5 กราฟแสดงความเร็วในการส่งข้อมูลของเครือข่ายของ Quadrics



รูปที่ 1.6 กราฟแสดงค่า Latency Time ของเครือข่ายของ Quadrics

- InfiniBand (http://www.infinibandta.org) เป็นเทคโนโลยีที่มีความเร็วในการสื่อสารข้อมูลสูงมากถึง 5 Gbit/sec สำหรับการ์ดรุ่น 1x และ 20 Gbit/sec สำหรับการ์ดรุ่นที่มีความเร็ว 4x อีกทั้งค่า Latency Time มีค่าต่ำมาก น้อยกว่า 10 microsecond

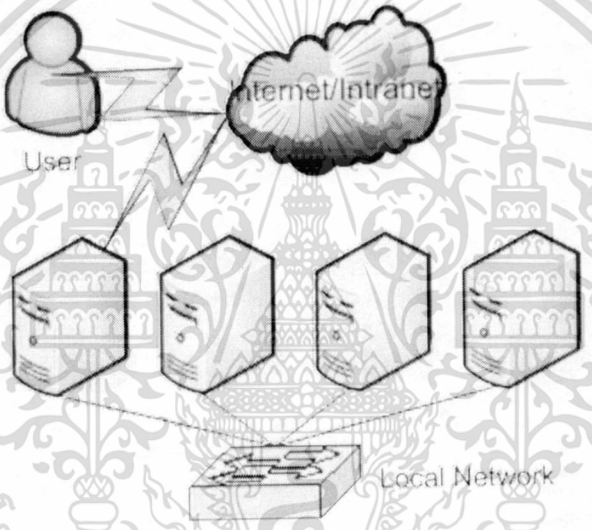
สำหรับรูปแบบวิธีการเชื่อมต่อเครือข่าย (Network Topology) ของระบบคลัสเตอร์นั้นมีหลายรูปแบบที่ถูกนำเสนอไว้ ไม่ว่าจะเป็นการเชื่อมต่อแบบวงแหวน การเชื่อมต่อแบบ cube และ HyperCube (Lester, 1993) ซึ่งรูปแบบการเชื่อมต่อเหล่านี้ต้องมีสายสัญญาณเชื่อมต่อระหว่างโหนดค่อนข้างมาก ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ต้องมีการลงทุนทางด้านเครือข่ายค่อนข้างสูง ดังนั้นรูปแบบของเครือข่ายแบบนี้จะไม่เหมาะกับการนำไปใช้งานในระบบจริง ดังนั้นในงานวิจัยนี้จึงนำเสนอเพียงแค่สองรูปแบบที่เป็นไปได้ในทางปฏิบัติและไม่ต้องลงทุนเกี่ยวกับเครือข่ายมากนัก ดังนี้

- การเชื่อมต่อโดยผ่านเกตเวย์

การเชื่อมต่อคลัสเตอร์ด้วยวิธีนี้จะเป็นดังภาพที่ 5.7 โดยจะมีคอมพิวเตอร์เครื่องหนึ่งที่ทำหน้าที่เป็นทางเข้าหรือทางออกหรือเกตเวย์ (Gate way) ให้แก่ระบบทั้งหมด เมื่อผู้ใช้ติดต่อเข้ามาในระบบจะต้องทำการติดต่อกับเครื่องที่เป็นเกตเวย์ ส่วนเครื่องอื่น ๆ จะทำงานอยู่เบื้องหลังเท่านั้น ทำให้วิธีนี้มีประโยชน์ในเรื่องของการรักษาความปลอดภัยของระบบ เพราะมีเพียงเครื่องเดียวเท่านั้นที่เชื่อมต่อกับภายนอก เมื่อต้องการรักษาความปลอดภัยก็สนใจเพียงเครื่องที่เป็นทางเข้าของระบบ แต่การเชื่อมต่อวิธีนี้ก็รับประกันไม่ได้เสมอไปนักว่าจะไม่ถูกบุกรุกจากภายนอก การเชื่อมต่อสามารถแสดงได้ดังนี้



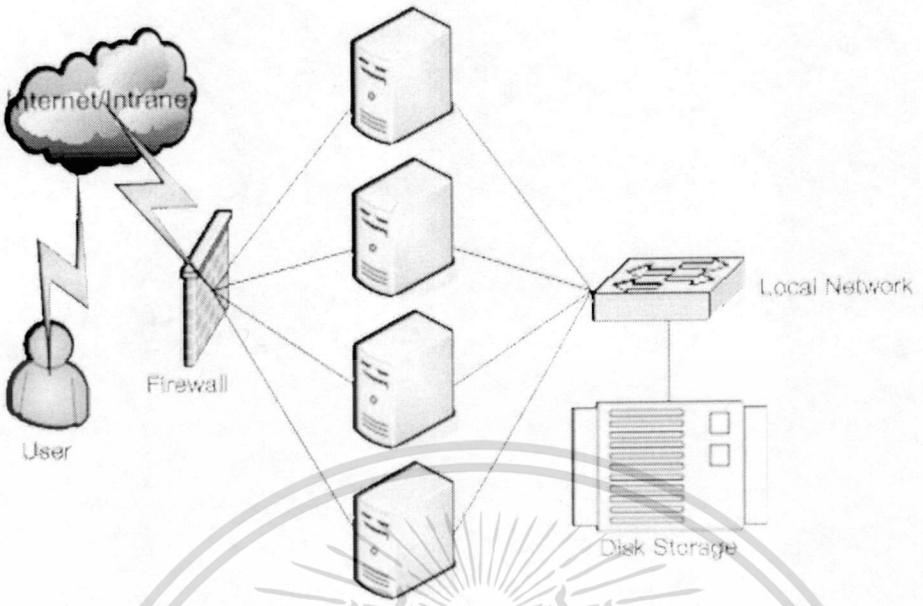
รูปที่ 1.7 การเชื่อมต่อกับเครือข่ายภายนอกผ่านเกตเวย์

การเชื่อมต่อแบบนี้มีข้อดีในส่วนของการดูแลเรื่องความปลอดภัย แต่จะไม่ค่อยเหมาะนักกับงานที่เป็นด้านการให้บริการหรือคลัสเตอร์ที่ต้องการความคงทนสูง (High Availability Cluster) อย่างเช่นเครื่องแม่ข่ายให้บริการเว็บ หรืออีเมล์ ดังนั้นจึงต้องใช้รูปแบบการเชื่อมต่อตามวิธีการในหัวข้อถัดไป

- การเชื่อมต่อแบบทุกเครื่องเชื่อมกับภายนอก

การเชื่อมต่อเครือข่ายแบบนี้ ทุกเครื่องจะทำการเชื่อมต่อกับเครือข่ายภายนอก ทำให้เครื่องจากภายนอกสามารถเข้าถึงทรัพยากรของแต่ละเครื่องได้โดยตรง โดยจะมีประโยชน์ในกรณีที่เครื่อง คลัสเตอร์เหล่านี้ทำหน้าที่ให้บริการงานต่าง ๆ เช่นเครื่องแม่ข่ายของงานเว็บไซต์ การเชื่อมต่อแบบนี้มักจะนำเอาไฟร์วอล (Firewall) มาวางไว้ด้านหน้าของระบบอีกชั้น เพื่อเป็นการป้องกันการบุกรุกจากภายนอกได้ในระดับหนึ่ง ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.8 การเชื่อมต่อแบบทุกเครื่องเชื่อมต่อกับเครือข่ายภายนอก

และเนื่องจากระบบนี้มักจะถูกใช้เป็นเครื่องแม่ข่ายที่ให้บริการอันเดียวกัน ดังนั้นจึงต้องมีที่เก็บข้อมูลรวมศูนย์ไว้ที่เดียวโดยอาจจะเป็น Storage ตามมาตรฐานทั่วไป เช่น NAS (Network Attach Server) เป็นต้น

สำหรับโครงสร้างพื้นฐานของระบบคลัสเตอร์มีทั้งหมดสี่ส่วนคือ โครงสร้างทางด้านฮาร์ดแวร์ ระบบปฏิบัติการ คลัสเตอร์มิดเดิลแวร์ และการเชื่อมต่อเครือข่าย ซึ่งได้กล่าวถึงโดยละเอียดแล้วในหัวข้อนี้ แต่ว่าการที่ระบบคลัสเตอร์จะทำงานได้อย่างมีประสิทธิภาพมากขึ้น หรือการที่จะทำให้ผู้ใช้งานมีความสะดวกในการเขียนโปรแกรมเพื่อทำงานบนระบบเหล่านี้ได้นั้น ต้องมีการกล่าวถึงรายละเอียดเพิ่มเติมของส่วนที่เป็นชุดคำสั่งทางซอฟต์แวร์ (Software Library) และเครื่องมือต่าง ๆ ดังในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 เครื่องมือและโปรแกรมอัตถะประโยชน์

ในหัวข้อนี้จะเป็นการกล่าวถึงเครื่องมือต่าง ๆ ที่ถูกใช้ในการสร้างและการเขียนโปรแกรมบนคอมพิวเตอร์ระบบคลัสเตอร์ โดยจะกล่าวถึงโปรแกรมจัดการสภาวะการโปรแกรมแบบขนาน หรือ Parallel Programming Environment ซึ่งในระบบคลัสเตอร์ส่วนใหญ่มักจะเลือกใช้โปรแกรมการสื่อสารระหว่างโหนดโดยการส่งข้อความหรือ Message Passing Interface นั่นเอง หลังจากนั้นจะเป็นการกล่าวถึงตัวแปรภาษาที่สนับสนุนการทำงานบนระบบคลัสเตอร์ และสุดท้ายจะได้กล่าวถึงชุดคำสั่งทางคณิตศาสตร์ที่ถูกพัฒนาขึ้นมาเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ให้สามารถทำได้ง่ายและสะดวกสบายมากยิ่งขึ้น ดังมีรายละเอียดต่าง ๆ ต่อไปนี้

1.7.1 โปรแกรมสื่อสารระหว่างโหนดโดยการส่งข้อความ

ในรูปที่ 5.3 ซึ่งแสดงส่วนประกอบทั้งหมดของระบบคลัสเตอร์ ในชั้นของสภาวะแวดล้อมการโปรแกรมแบบขนาน (Parallel Programming Environment) นั้น จะเป็นชั้นของซอฟร์แวร์ที่จะคอยอำนวยความสะดวกพื้นฐานให้แก่ผู้ใช้งานระบบ ทั้งในส่วนของชุดคำสั่ง (Library) พื้นฐานที่จำเป็นในการสื่อสารระหว่างโหนด การส่งค่าตัวแปร การส่งค่าข้อมูลแบบเจาะจง การส่งข้อมูลแบบกระจายหรือบรอดแคสต์ (Broadcasting) และอื่น ๆ อีกทั้งมีการเตรียมซอฟต์แวร์สำหรับการติดต่อ (Software Interface) กับตัวแปรภาษาหรือคอมไพเลอร์ และคำสั่งใช้งานทั่วไป

สำหรับมาตรฐานของสภาวะแวดล้อมการโปรแกรมแบบขนานที่ได้รับความนิยม และมีการพัฒนาชุดคำสั่งตามมาตรฐานขึ้นมาใช้งานจริงมากที่สุดคือ มาตรฐานการส่งผ่านข้อความ หรือ Message Passing Interface ซึ่งมีชุดคำสั่งที่สร้างขึ้นมาใช้งานและได้รับความนิยมมากคือ MPICH (http://www-unix.mcs.anl.gov/mpi/mpich/) และ LAM (http://www.lam-mpi.org)

โดยทั้ง MPICH และ LAM จะเป็นซอฟแวร์ไลบรารี (Software Library) ที่ถูกพัฒนาขึ้นมาตามมาตรฐาน MPI (http://www.mpi-forum.org/) เวอร์ชั่น 1.1 หรือสูงกว่า และจะมีการเตรียมฟังก์ชั่นพื้นฐานต่าง ๆ มากมาย รวมทั้งมีคำสั่งใช้งานพื้นฐานให้แก่ผู้ใช้เพื่อทำการคอมไพล์ (Compile) โปรแกรมที่เขียนขึ้นมาและคำสั่งใช้งานอื่น ๆ เพื่อติดต่อและสั่งงานกับระบบคลัสเตอร์

ซึ่งรายละเอียดเกี่ยวกับรายละเอียดของการเขียนโปรแกรมแบบขนาน และการใช้งานคำสั่งของ MPICH นั้น จะกล่าวโดยละเอียดในบทถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7.2 คอมไพเลอร์

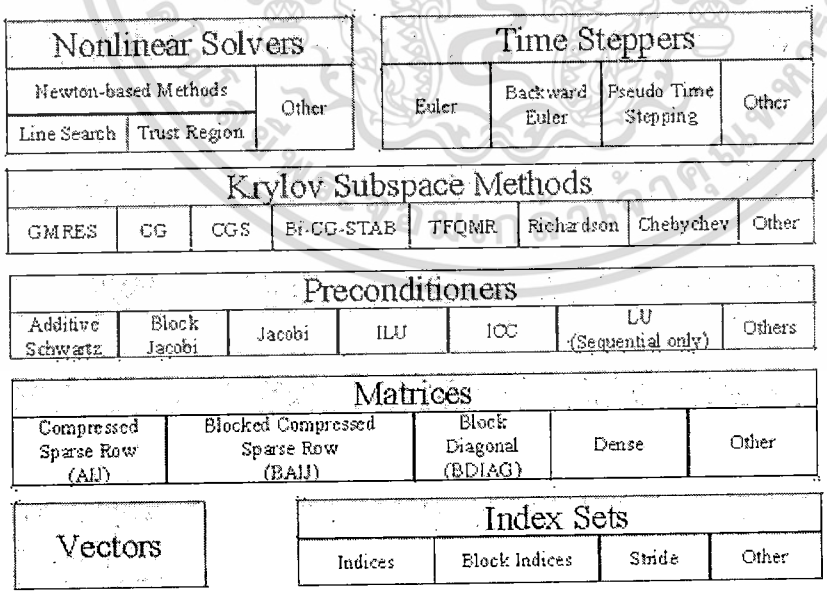
ภาษาคอมพิวเตอร์ที่ถูกนำมาใช้ในการเขียนโปรแกรมบนระบบคลัสเตอร์ ที่ได้รับความนิยมโดยส่วนใหญ่คือ C/C++ และ FORTRAN เพราะเนื่องจากมีผู้พัฒนาฟังก์ชั่นและไลบรารีต่าง ๆ ขึ้นมากมาย ทั้งตัว MPICH, LAM หรือ PVM (Parallel Virtual Machine) เอง ก็มีฟังก์ชั่นสนับสนุนภาษาเหล่านี้โดยตรง

เมื่อจะทำการสร้างโปรแกรมขนานจะต้องมีการคอมไพล์โปรแกรมที่สร้างขึ้นมา และเนื่องจากฟังก์ชั่นที่ถูกเตรียมไว้ให้โดย MPICH หรือ LAM เป็น C/C++ หรือ FORTRAN ดังนั้นในระบบคลัสเตอร์จึงต้องมีตัวแปรภาษาเหล่านี้อยู่ในระบบก่อนแล้ว คอมไพเลอร์ที่นิยมนำมาใช้ตัวอย่างเช่น GCC ของ GNU Project (http://gcc.gnu.org) เป็นต้น

1.7.3 ชุดคำสั่งทางคณิตศาสตร์

มีผู้พัฒนาชุดคำสั่งทางคณิตศาสตร์ขึ้นมาเพื่อความสะดวกในการสร้างโปรแกรมการคำนวณต่าง ๆ หลายอย่าง ซึ่งชุดคำสั่งเหล่านี้ได้ช่วยให้การสร้างโปรแกรมขึ้นมาทำได้ง่ายขึ้นเพราะไม่ต้องเขียนโปรแกรมเองทั้งหมด และคำสั่งเหล่านี้ก็สามารถใช้งานร่วมกับระบบคลัสเตอร์ได้โดยไม่มีปัญหา และบางชุดคำสั่งยังสนับสนุนการทำงานแบบขนานในตัวชุดคำสั่งเลยอีกด้วย ตัวอย่างชุดคำสั่งเหล่านี้เช่น

- PETSc (http://acts.nersc.gov/petsc/) เป็นไลบรารีที่ถูกสร้างขึ้นมาเพื่อแก้ปัญหาทางด้านนิวเมอริคอล (Numerical Analysis) โดยเฉพาะสมการพวก Partial differential equation สมการเชิงเส้น และสมการไม่เชิงเส้น เป็นต้น



รูปที่ 1.9 ส่วนประกอบของชุดคำสั่ง PETSc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PLAPACK (http://www.cs.utexas.edu/users/plapack/) หรือ Parallel Linear Algebra Package เป็นชุดคำสั่งที่สนับสนุนการประมวลผลแบบขนาน เพราะถูกสร้างขึ้นมาบนพื้นฐานของ MPI และชุดคำสั่งนี้ถูกสร้างขึ้นมาเพื่อการแก้ปัญหาของสมการพีชคณิตเชิงเส้น และการคำนวณเกี่ยวกับเมตริกซ์
- ScaLAPACK (http://www.netlib.org/scalapack/) เป็นชุดคำสั่งที่ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาเกี่ยวกับการคำนวณเมตริกซ์ และพีชคณิตเชิงเส้น การหาค่า eigenvalue เป็นต้น

1.8 การจัดการระบบคลัสเตอร์

ระบบคลัสเตอร์ที่ดีนั้นนอกจากจะทำงานได้อย่างมีประสิทธิภาพ และคำนวณงานต่าง ๆ ได้อย่างรวดเร็วแล้ว จะต้องมีการจัดการที่ดี สามารถรองรับการทำงานของผู้ใช้งานหลาย ๆ คน รวมทั้งสามารถตรวจสอบสถานะของแต่ละโหนดได้อีกด้วย การจัดการระบบที่ดีจะทำให้เกิดความสะดวกแก่ผู้ใช้งาน โดยส่วนใหญ่รวมทั้งทำให้เกิดประสิทธิภาพในการทำงานมากขึ้นด้วย การจัดการระบบนั้นจะมีอยู่ด้วยกันสามส่วนคือ

- **การจัดการเกี่ยวกับผู้ใช้** (Users Management) จะเป็นการเพิ่มหรือลบผู้ใช้ในระบบ ให้สิทธิ์ต่าง ๆ ที่เหมาะสมกับผู้ใช้งานแต่ละคน กำหนดความเหมาะสมให้เกิดขึ้นสำหรับผู้ใช้งาน และเนื่องมาจากเหตุผลที่ว่าคอมพิวเตอร์แต่ละโหนดที่เชื่อมต่ออยู่ในคลัสเตอร์นั้นสามารถทำงานได้โดยอิสระ และมีระบบปฏิบัติการเป็นของตัวเอง การจัดการเรื่องผู้ใช้งานจึงเป็นสิ่งจำเป็นอันดับต้น ๆ เพราะเมื่อทำการเพิ่มหรือลบผู้ใช้ออกจากระบบ ทุกโหนดจะต้องรู้และรับทราบการเปลี่ยนแปลงนั้น อีกอย่างเมื่อผู้ใช้ทำการล็อกอินเข้ามายังระบบคงไม่ดีแน่หากผู้ใช้ต้องทำการล็อกอินไปยังทุกเครื่องก่อนจึงจะสามารถใช้งานได้ ดังนั้นทางที่ดีควรจะทำให้ผู้ใช้ล็อกอินเข้าสู่ระบบเพียงครั้งเดียวแล้วสามารถใช้ทรัพยากรภายในระบบได้อย่างเหมาะสม
- **การจัดการคิวงาน** (Jobs scheduler) เมื่อมีผู้ใช้งานระบบหลาย ๆ คน อาจจะมีการแย่งกันใช้ทรัพยากรของระบบที่มีอยู่ ดังนั้นจะต้องมีระบบหรือวิธีการที่จะมาเรียงลำดับหรือช่วยจัดการงานต่าง ๆ ตามความสำคัญหรือตามลำดับของงานที่ส่งให้มาประมวลผลในระบบ ระบบการจัดการคิวงานที่ดีนั้นต้องสามารถตรวจสอบทรัพยากรที่มีอยู่ของระบบทั้งหมดได้ เช่น มีโหนดออนไลน์ (Online) อยู่กี่โหนด หรือออฟไลน์ (Offline) ไปแล้วกี่โหนด ผู้ใช้คนไหนมีสิทธิ์ใช้หน่วยประมวลผลได้มากน้อยแค่ไหน ใช้หน่วยความจำได้เท่าไหร่ โปรแกรมของผู้ใช้สามารถทำงานได้ตั้งแต่เวลาเท่าไหร่ถึงเท่าไหร่ ซึ่งสิ่งเหล่านี้ล้วนแต่เป็นสิ่งที่ต้องมีในตัวจัดการคิวงาน ตัวจัดการคิวงานนี้มีเครื่องมือให้ใช้บ้างบางส่วนนั่นคือ ระบบ Batch queues ซึ่งที่นิยมนำมาใช้งานเช่น OpenPBS (Open Portable Batch System)
- **การตรวจสอบและการจัดการ** (Monitoring and Management) ระบบคลัสเตอร์จะต้องทำการตรวจสอบสถานะทำงานได้ เช่น สถานะการใช้งานหน่วยประมวลผล ปริมาณการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ และสถานะของเครื่องแต่ละโหนด เพื่อจะได้ทำการตรวจสอบและแก้ไขได้อย่างถูกต้อง ระบบการตรวจสอบสถานะมีเครื่องมือ เช่น Ganglia (http://ganglia.sourceforge.net/) เป็นต้น ซึ่งเครื่องมือเหล่านี้ทำให้จัดการระบบได้ง่ายขึ้น เพราะมีการตรวจสอบระบบเป็นระยะ ส่วนระบบการจัดการนั้นต้องมีระบบการจัดการที่ดีเพื่อให้ระบบสามารถทำงานต่อเนื่องยาวนาน และมีประสิทธิภาพสูงสุด

1.9 สรุป

ในบทนี้ได้กล่าวถึงความรู้เบื้องต้นเกี่ยวกับการประมวลผลแบบขนาน รูปแบบและโครงสร้างของการประมวลผลแบบขนาน ทั้งนี้ได้กล่าวถึงรายละเอียดของคอมพิวเตอร์ความเร็วสูงแบบคลัสเตอร์ กริดคอมพิวเตอร์ และเครือข่ายคอมพิวเตอร์ประมวลผลแบบ P2P แล้วค่อยเจาะลึกลงในรายละเอียดของส่วนประกอบต่าง ๆ ของระบบคลัสเตอร์ ทั้งในส่วนของโครงสร้างพื้นฐานของระบบคลัสเตอร์ เครื่องมือและโปรแกรมอัตถะประโยชน์ต่าง ๆ และสุดท้ายคือการจัดการระบบให้ทำงานได้อย่างมีประสิทธิภาพ สามารถรองรับการใช้งานจากผู้ใช้จำนวนมาก ๆ ได้ ตลอดทั้งมีวิธีการที่ทำให้สามารถตรวจสอบระบบได้ง่ายขึ้นด้วย

ตลอดทั้งบทนี้ได้แนะนำให้เห็นรายละเอียดว่าระบบคลัสเตอร์ประกอบด้วยส่วนประกอบใดบ้าง และมีเครื่องมือและวิธีการในการจัดการระบบอย่างไร ส่วนในบทต่อไปจะเป็นการแนะนำให้รู้จักถึงขั้นตอนและวิธีการของการโปรแกรมแบบขนาน และการทำงานบนระบบคลัสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
โปรแกรมแบบขนาน

2.1 บทนำ

หลักการประมวลผลขนานแบบต่าง ๆ ทั้งระบบที่เป็นแบบหลายหน่วยประมวลผลหรือหลายเครื่องประมวลผล จะไม่สามารถทำงานได้อย่างมีประสิทธิภาพ ถ้าหากโปรแกรมที่นำมาใช้งานบนระบบเหล่านี้ไม่ได้ทำการออกแบบและสร้างขึ้นมาอย่างเหมาะสมหรือยังคงใช้วิธีการในการโปรแกรมแบบลำดับขั้น (Sequential Algorithm) เหมือนกับโปรแกรมอื่น ๆ ทั่วไป ดังนั้นในบทนี้จึงจะเป็นการกล่าวถึงความสำคัญและวิธีการในการออกแบบและสร้างโปรแกรมแบบขนาน โดยใช้เทคนิคและวิธีการต่าง ๆ เพื่อให้เกิดประสิทธิภาพในการทำงานของโปรแกรมมากยิ่งขึ้น

ในบทนี้จะเป็นการกล่าวถึงเนื้อหาทั้งหมดของการโปรแกรมแบบขนาน นับตั้งแต่กลวิธีที่ถูกนำมาใช้ในการโปรแกรม รูปแบบและวิธีการพัฒนาโปรแกรม การออกแบบวิธีการการทำงานของโปรแกรม ทั้งการแบ่งงาน การรวมกลุ่มงาน การสมดุลงาน และการสื่อสารระหว่างโปรเซสในขณะทำงาน หลังจากนั้นจะเป็นการกล่าวถึงการออกแบบวิธีการสร้างภาพเชิงปริมาตรทางการแพทย์บนระบบคลัสเตอร์โดยใช้การโปรแกรมแบบขนาน

2.2 รูปแบบของการพัฒนาโปรแกรมแบบขนาน

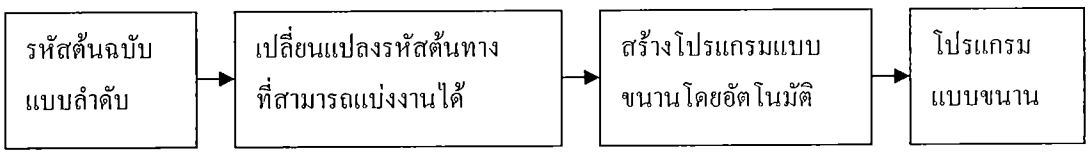
ในหัวข้อนี้จะเป็นการกล่าวถึงรูปแบบหรือแนวทางในการพัฒนาโปรแกรมขนานแบบต่าง ๆ ก่อนที่จะกล่าวถึงทฤษฎีและการออกแบบเพื่อสร้างโปรแกรมขนานในหัวข้อถัดไป โดยลำดับแรกในการพัฒนาโปรแกรมขนานนั้น ส่วนใหญ่ผู้เขียนโปรแกรมมักจะสร้างโปรแกรมในแบบลำดับ (Sequential Program) ให้สามารถทำงานได้อย่างถูกต้องก่อน จากนั้นจึงจะพัฒนาโปรแกรมแบบลำดับนี้ไปเป็นโปรแกรมขนานได้ตามวิธีเหล่านี้

2.2.1 การสร้างโปรแกรมแบบขนานโดยอัตโนมัติ

การสร้างโปรแกรมด้วยวิธีนี้เป็นวิธีที่ง่ายที่สุด แต่ก็จะมีประสิทธิภาพต่ำที่สุด โดยหน้าที่ของการสร้างโปรแกรมแบบขนานจะเป็นหน้าที่ของตัวแปรภาษาซึ่งตัวแปรภาษาจะตรวจสอบรหัสต้นฉบับ (Source Code) ซึ่งอาจประกอบด้วยส่วนของรหัสที่มีการวนซ้ำ และการประมวลผลกับข้อมูลที่ซ้ำๆกัน โดยตัวแปรภาษานี้จะทำการเปลี่ยนรหัสต้นฉบับไปเป็นรหัสที่สนับสนุนการประมวลผลแบบขนาน จากนั้นใช้ตัวแปรภาษาเปลี่ยนรหัสที่ได้ไปเป็นโปรแกรมแบบขนานเองโดยอัตโนมัติ แต่ข้อจำกัดของการสร้างโปรแกรมด้วยวิธีนี้จะขึ้นอยู่กับเทคโนโลยีที่ตัวแปรภาษาใช้ในการเปลี่ยนรหัสต้นฉบับเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสที่สนับสนุนการประมวลผลแบบขนาน ขั้นตอนการทำงานของการสร้างโปรแกรมขนานโดยอัตโนมัติแสดงดังรูปที่ 6.1



รูปที่ 2.1 ขั้นตอนการทำงานของการสร้างโปรแกรมแบบขนานโดยอัตโนมัติ

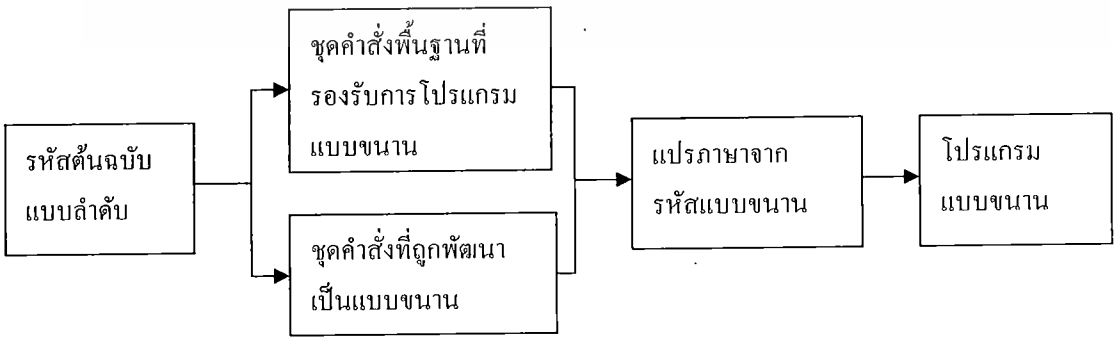
2.2.2 การสร้างโปรแกรมโดยใช้ชุดคำสั่งแบบขนาน

เป็นวิธีการที่มีประสิทธิภาพมากกว่าวิธีแรก โดยเมื่อทำการสร้างโปรแกรมจะใช้ชุดคำสั่งที่ถูกพัฒนาขึ้นมาให้สนับสนุนการทำงานแบบขนานแล้ว เช่น ชุดคำสั่งของ ScaLAPACK หรือ PLAPACK ซึ่งชุดคำสั่งเหล่านั้นสนับสนุนการประมวลผลแบบขนานอยู่แล้ว ฟังก์ชั่นที่ชุดคำสั่งเหล่านี้เตรียมไว้ให้ใช้งาน เช่น ชุดคำสั่งของการหาผลเฉลยสมการต่าง ๆ แบบขนาน และการคูณเมตริกแบบขนาน เป็นต้น

ชุดคำสั่งที่สนับสนุนการทำงานแบบขนานนี้จะมีอยู่สองกลุ่มคือ

- กลุ่มของชุดคำสั่งพื้นที่รองรับหรือทำให้เกิดสภาพแวดล้อมการประมวลผลแบบขนาน ตัวอย่างเช่น ฟังก์ชั่นที่ถูกเตรียมไว้โดย MPICH หรือ LAM ซึ่งจะเป็นคำสั่งหรือฟังก์ชั่นพื้นฐานที่ทำให้เกิดการประมวลผลแบบขนานได้ เพราะมีคำสั่งที่ทำให้เกิดการสื่อสารระหว่างโหนดต่อโหนด การส่งข้อความถึงโหนดอื่น ๆ ในระบบได้ และอื่น ๆ อีกมาก
- กลุ่มของชุดคำสั่งที่ได้รับการพัฒนาให้เป็นการประมวลผลแบบขนาน เช่น ScaLAPACK หรือ PLAPACK ซึ่งเป็นชุดคำสั่งทางคณิตศาสตร์ที่ได้รับการพัฒนาให้ทำงานแบบขนานได้โดยเรียกใช้ชุดคำสั่งของ MPI อีกที

ในรายการวิจัยฉบับนี้ใช้วิธีการสร้างโปรแกรมแบบขนานโดยใช้เพียงแค่คำสั่งพื้นฐานเพื่อให้เกิดสภาพแวดล้อมในการโปรแกรมแบบขนานขึ้นเท่านั้น แต่ไม่ได้ใช้ชุดคำสั่งขนานแบบอื่นเข้ามาช่วย

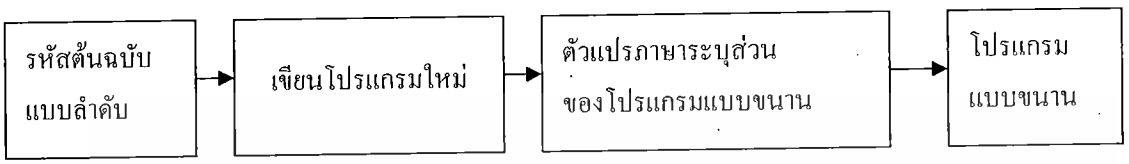


รูปที่ 2.2 ขั้นตอนการสร้างโปรแกรมโดยใช้ชุดคำสั่งแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การสร้างโปรแกรมแบบขนานด้วยตัวเอง

การสร้างโปรแกรมแบบขนานด้วยวิธีนี้เป็นแบบที่ยืดหยุ่นมากที่สุด ผู้สร้างโปรแกรมสามารถเลือกตัวแปรภาษาใดก็ได้ที่ต้องการ รวมถึงสามารถเลือกรูปแบบและวิธีการที่ใช้ในการสื่อสาร แต่วิธีนี้นับเป็นวิธีที่ยากและเสียเวลามากที่สุด ขั้นตอนการสร้างโปรแกรมแบบขนานด้วยตัวเองสามารถแสดงได้ดังรูปที่ 6.3



รูปที่ 2.3 ขั้นตอนการสร้างโปรแกรมแบบขนานด้วยตัวเอง

2.3 กลวิธีการโปรแกรมแบบขนาน

การสร้างโปรแกรมแบบลำดับขั้น (Sequential Algorithm) บนเครื่องคอมพิวเตอร์ที่มีเพียงหน่วยประมวลผลเดียวเป็นเรื่องที่ไม่ยุ่งยากหรือสลับซับซ้อนมากมายนัก เพราะลำดับการทำงานของโปรแกรมเป็นไปอย่างตรงไปตรงมากับความต้องการหรือทฤษฎีและวิธีการที่อธิบายไว้แล้ว การทำงานของโปรแกรมใช้เพียงหน่วยประมวลเดียว การเข้าใช้งานหน่วยความจำก็สามารถที่เข้าถึงและใช้งานได้โดยไม่มีปัจจัยอย่างอื่นเข้ามาเกี่ยวข้อง การสื่อสารระหว่างโปรเซสที่ทำงานอยู่ก็มักจะไม่เกิดขึ้นในวิธีการการโปรแกรมแบบนี้ ทำให้การออกแบบโปรแกรมสามารถทำได้ง่าย ซึ่งแตกต่างจากกลวิธีการสร้างโปรแกรมแบบขนาน เพราะปัจจัยในเรื่องการใช้งานหน่วยประมวล การเข้าใช้งานหน่วยความจำและการสื่อสารระหว่างโปรเซสจะเริ่มเข้ามามีบทบาทในการทำงานของโปรแกรมมากขึ้น อีกทั้งถ้าหากต้องการให้โปรแกรมทำงานได้อย่างมีประสิทธิภาพแล้วต้องคำนึงถึงปัจจัยอื่น ๆ ที่มาเกี่ยวข้องด้วย

ดังนั้นในการสร้างโปรแกรมแบบขนานจึงได้มีผู้นำเสนอเทคนิคและวิธีการต่าง ๆ หลายอย่างในการสร้างโปรแกรมขึ้นมา เพื่อให้การทำงานของโปรแกรมเป็นไปอย่างเหมาะสมและมีประสิทธิภาพ Lester (1993:13) ได้นำเสนอวิธีการต่าง ๆ โดยรวมไว้ทั้งหมด 6 วิธีการดังนี้

- **การรวมกลุ่มชุดข้อมูลที่มีความเหมือนกัน** (Data Parallelism) ซึ่งเป็นวิธีการที่สามารถใช้งานได้โดยทั่วไป หลักการของการสร้างโปรแกรมแบบนี้ จะทำการรวมเอาข้อมูลอินพุท (Input) ที่มีความเหมือนกันเข้าไว้ด้วยกันเป็นกลุ่ม หลังจากนั้นทำการประมวลผลแยกเป็นกลุ่ม ๆ ไป เมื่อข้อมูลที่อยู่ในกลุ่มเดียวกันมีความสัมพันธ์กันแล้ว ทำให้เกิดความง่ายในการคำนวณมากยิ่งขึ้น
- **การแบ่งกลุ่มข้อมูลที่อยู่ใกล้กันไว้ด้วยกัน** (Data Partitioning) ข้อมูลอินพุทที่อยู่ใกล้กันจะมีความสัมพันธ์กันอยู่แล้วส่วนหนึ่ง ดังนั้นจึงสามารถที่จะรวมกลุ่มชุดของข้อมูลที่อยู่ใกล้กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไว้ด้วยกันได้ วิธีนี้ทำให้เกิดความสะดวกในการแบ่งกลุ่มของข้อมูลมากยิ่งขึ้น เพราะเพียงพิจารณาข้อมูลที่อยู่ติดกัน แยกเป็นส่วน ๆ เท่านั้น เมื่อทำการแยกข้อมูลได้แล้วก็ทำการประมวลผลข้อมูลกลุ่มต่าง ๆ ไปพร้อมกันตามหลักการของการประมวลผลแบบขนาน

วิธีการนี้เหมาะกับการคำนวณบนระบบแบบหลายเครื่องประมวลผลอย่างเช่นระบบคลัสเตอร์เป็นอย่างมาก เพราะแต่ละเครื่องจะมีหน่วยความจำเป็นของตัวเองและเมื่อทำการคำนวณแต่ละเครื่องหรือแต่ละโหนดจะเกี่ยวข้องกับชุดข้อมูลที่ถูกแบ่งออกมาอย่างชัดเจน การเกี่ยวข้องกันกับชุดข้อมูลข้างเคียงมีน้อย ทำให้เกิดการแลกเปลี่ยนข้อมูลระหว่างโหนดน้อยลงตามไปด้วย มีผลทำให้เวลาที่สูญเสียไปในการสื่อสารน้อย เวลารวมในการทำงานก็ลดลงด้วย

- **การทำงานแยกกันโดยอิสระ** (Relaxed Algorithm) ถ้าหากแต่ละหน่วยประมวลผลในระบบสามารถทำงานแยกจากกันได้โดยอิสระ โดยที่ไม่ต้องรอข้อมูลผลลัพธ์จากหน่วยประมวลผลหรือโปรเซสข้างเคียง และไม่ต้องสื่อสารหรือตรวจทานความถูกต้องกับหน่วยประมวลผลอื่น ๆ ทำให้ไม่ต้องสูญเสียเวลาในการสื่อสารหรือการรอในระหว่างการทำงานเลย หลักการนี้สามารถนำไปประยุกต์ใช้ได้ทั้งระบบแบบหลายหน่วยประมวลและหลายเครื่องประมวลผล
- **การทำงานพร้อมกันในแต่ละรอบการทำงาน** (Synchronous Iteration) ในการคำนวณบางประเภทอาจจะต้องมีการทำงานไปพร้อม ๆ กันในแต่ละรอบการทำงาน วิธีการนี้จะให้แต่ละโปรเซสทำการตรวจสอบการทำงานของโปรเซสข้างเคียงอื่น ๆ ในแต่ละรอบเพื่อให้คำตอบที่ได้มีค่าถูกต้อง การทำงานไปพร้อม ๆ กันจะลดความเร็วในการประมวลผลลงเล็กน้อย และเหมาะกับระบบที่เป็นแบบหลายหน่วยประมวลผลมากกว่าระบบที่เป็นแบบหลายเครื่องประมวลผล เพราะจะต้องมีการสื่อสารระหว่างกันค่อนข้างมาก
- **การกระจายการทำงาน** วิธีการนี้จะทำการรวมศูนย์งานไว้ที่เดียวเรียกว่า Work Pool เครื่องอื่น ๆ หรือโหนดอื่น ๆ ในระบบจะถูกมองว่าเป็นผู้ทำงานหรือ Worker ซึ่งผู้ทำงานนี้จะต้องเข้ามารับงานจาก Work Pool ไปทำจนกระทั่งงานที่อยู่ใน Pool หมดลง การทำงานแบบนี้อาจกล่าวง่าย ๆ ว่าเป็นการทำงานแบบรวมศูนย์ เพราะทุก Worker จะทำการติดต่อและร้องของานจากที่เดียวกัน
- **การส่งต่อค่าผลลัพธ์** (Pipelined Computation) เทคนิคนี้ได้อาศัยรูปแบบของการส่งต่อค่าข้อมูลผลลัพธ์จากโปรเซสหนึ่ง ไปยังอีกโปรเซสหนึ่งในระบบเป็นลำดับเรื่อยไป ซึ่งการคำนวณงานบางชนิดต้องใช้รูปแบบการคำนวณงานแบบนี้ การส่งต่อค่าผลลัพธ์จะสามารถใช้ได้ดีทั้งระบบแบบหลายหน่วยประมวลผลและระบบแบบหลายเครื่องประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้โปรแกรมทำงานอย่างมีประสิทธิภาพสูงสุด เมื่อจะทำการออกแบบโปรแกรมนั้น Lester (1993:15) ได้เสนอถึงปัญหาต่าง ๆ ที่ควรคำนึงถึง รวมทั้งปัญหาที่ควรหลีกเลี่ยงไม่ให้เกิดขึ้นในโปรแกรม ปัญหาสำคัญ ๆ มีดังนี้

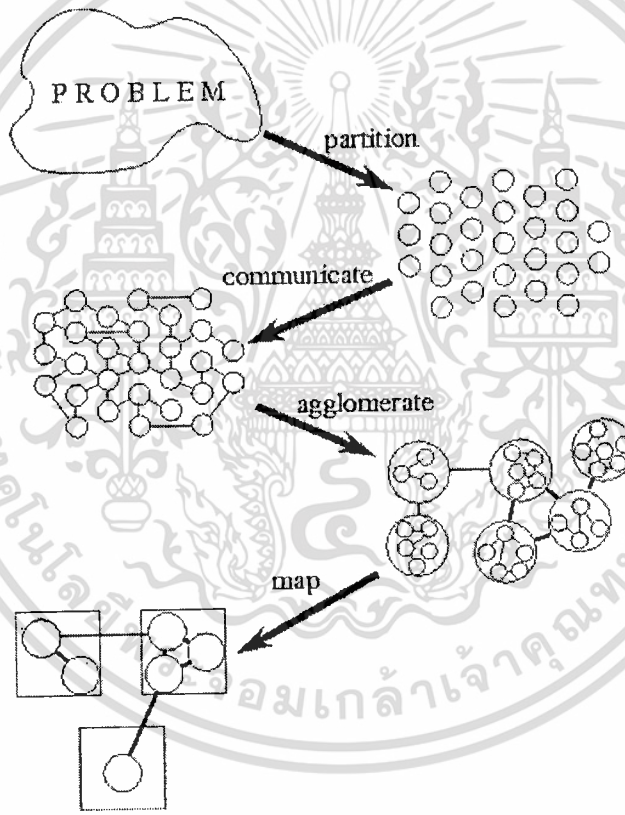
- ความคับคั่งในการเข้าใช้งานหน่วยความจำ (Memory Contention) ปัญหานี้เกิดจากการเข้าใช้ตัวแปรร่วม (Global variable) ในหน่วยความจำพร้อมกันของหน่วยประมวลผล เมื่อหน่วยประมวลผลหนึ่งใช้งานหน่วยความจำอยู่ หน่วยประมวลผลอื่น ๆ จะไม่สามารถเข้าใช้งานได้ ทำให้เกิดการหยุดรอของหน่วยประมวลผลขณะที่กำลังคำนวณ ซึ่งจะทำให้สูญเสียเวลาไปกับการรอนี้ แต่ปัญหานี้มักจะเกิดในระบบหลายหน่วยประมวลผลที่ใช้หน่วยความจำร่วมมากกว่าส่วนระบบแบบหลายเครื่องประมวลผลมักไม่เป็นปัญหา
- การมีส่วนของโปรแกรมที่ทำงานแบบลำดับมากเกินไป (Excessive Sequential Code) ซึ่งอาจจะเกิดจากตอนที่ออกแบบโปรแกรมนั้น ผู้เขียนโปรแกรมอาจจะสร้างโปรแกรมขึ้นมาจากวิธีการแบบลำดับ และยังใช้วิธีการคิดแบบเดิมในการโปรแกรมแบบขนานอยู่ ทำให้โปรแกรมที่ถูกสร้างขึ้นไม่ได้ทำงานไปอย่างพร้อม ๆ กันอย่างแท้จริง และประสิทธิภาพของโปรแกรมก็ลดลง
- การสูญเสียเวลาไปในการสร้างโปรเซส (Process Creation Time) ในการทำงานแบบขนานจะต้องมีเวลาส่วนหนึ่งที่สูญเสียไปเพราะการสร้างโปรเซสขึ้นมาใหม่ เพราะฉะนั้นการสร้างโปรเซสควรสร้างขึ้นมาเท่าที่จำเป็นเท่านั้น เพราะบางทีการสร้างโปรเซสขึ้นมามากเกินไปก็ไม่ได้ช่วยทำให้การทำงานของโปรแกรมเร็วขึ้นแต่อย่างใด
- การสูญเสียเวลาในการสื่อสาร (Communication Delay) ปัญหานี้จะเกิดขึ้นกับระบบแบบหลายเครื่องประมวลผลค่อนข้างมาก เพราะแต่ละหน่วยประมวลผลจะต้องทำการติดต่อกันผ่านทางการส่งข้อความข้ามไปมาระหว่างโปรเซสที่ทำงานอยู่ในระบบ ดังนั้นเมื่อทำการออกแบบโปรแกรมแล้วต้องให้มีการสื่อสารระหว่างโปรเซสน้อยที่สุด และขนาดของข้อมูลที่รับและส่งนั้นควรมีขนาดเล็กมากที่สุดด้วยเพื่อให้ เสียเวลาในการสื่อสารน้อยที่สุดนั่นเอง
- การสูญเสียเวลาในการตรวจทานสถานะ (Synchronization Delay) เมื่อมีการออกแบบโปรแกรมให้แต่ละโปรเซสต้องทำการตรวจทานสถานะซึ่งกันและกันแล้ว จะต้องมีเวลาส่วนหนึ่งสูญเสียไปเพื่อการนี้ ดังนั้น ทางที่ดีถ้าหากลดการตรวจทานสถานะไปได้หรือไม่มีเลยก็จะทำให้โปรแกรมที่ได้มีประสิทธิภาพดีขึ้นด้วย
- การกระจายงานไม่สมดุล หรือไม่เหมาะสม (Load Imbalance) เกิดจากการที่มีวิธีการกระจายการทำงานไม่เหมาะสม ทำให้มีหน่วยประมวลผลบางส่วนเท่านั้นที่มีโอกาสได้ทำงานส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยประมวลผลอื่นอาจจะอยู่ในสถานะรอเพียงอย่างเดียว ดังนั้นเพื่อแก้ปัญหานี้จะต้องมีวิธีการที่เหมาะสมในการกระจายงานไปยังหน่วยประมวลผลในระบบ

2.4 การออกแบบโปรแกรมขนาน

เมื่อจะทำการลงมือเขียนโปรแกรมนั้น จะต้องมีการออกแบบโปรแกรมที่ดีก่อนจึงค่อยลงมือเขียน โดยโปรแกรมที่ดีนั้นจะต้องเป็นไปตามหลักการและหลีกเลี่ยงปัญหาต่าง ๆ ที่ได้นำเสนอไปแล้วในหัวข้อที่ผ่านมา ซึ่งวิธีการออกแบบโปรแกรมขนานนี้ Foster (http://www-unix.mcs.anl.gov/dbpp/. 1995) ได้นำเสนอหลักการต่าง ๆ ไว้สี่ขั้นตอนด้วยกันคือ การแบ่งงาน (Partition) การพิจารณาเรื่องการสื่อสาร (Communication) การรวมกลุ่มงาน (Agglomerate) และการกำหนดงานไปยังหน่วยคำนวณที่เหมาะสม (Mapping) สามารถแสดงขั้นตอนทั้งหมดได้ดังรูปต่อไปนี้



รูปที่ 2.4 ขั้นตอนการออกแบบการทำงานของโปรแกรมขนาน

จากรูปจะเห็นได้ว่าจากปัญหาขนาดใหญ่ที่ต้องการคำนวณนั้น ในขั้นตอนของการแบ่งงานจะทำการแยกแยะปัญหาออกให้เป็นส่วนย่อยตามความเหมาะสมของแต่ละปัญหา หลังจากนั้นจึงทำการพิจารณาความเกี่ยวพันของแต่ละส่วน โดยอาจจะพิจารณาจากขอบเขต การอยู่ติดกันของส่วนย่อยเล็ก ๆ เหล่านั้น รวมทั้งพิจาณาถึงการรับและส่งค่าของปัญหาต่าง ๆ อีกด้วย หลังจากนั้นจึงทำการรวมส่วนที่

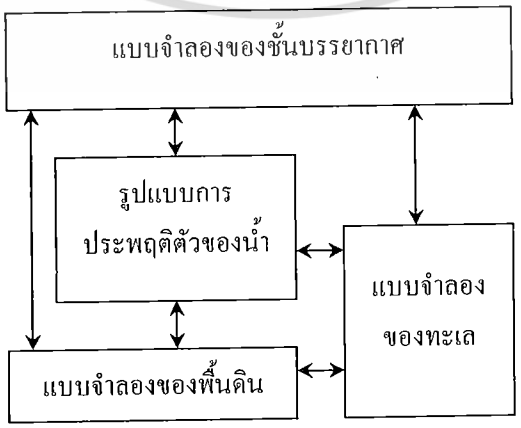
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกี่ยวข้องหรือมีความสัมพันธ์เข้าไว้ด้วยกันในขั้นตอนของการรวมกลุ่มงาน และสุดท้ายจะเป็นการกำหนดงานไปยังหน่วยคำนวณหรือ Processing Element ที่เหมาะสมต่อไป

2.4.1 ขั้นตอนการแบ่งงาน

การแบ่งงานเป็นจุดเริ่มต้นของการประมวลผลแบบขนาน เพราะจะต้องทำการแบ่งปัญหาขนาดใหญ่ที่ซับซ้อนออกมาเป็นส่วนย่อยที่สามารถคำนวณไปพร้อม ๆ กันให้ได้ก่อน จึงจะสามารถทำตามขั้นตอนอื่น ๆ ต่อไปได้ การแบ่งงานสามารถทำได้สองแบบคือ

- การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อย (Data Decomposition) ในส่วนนี้จะเป็นการพิจารณาที่ข้อมูลอินพุท ว่าสามารถแบ่งเป็นส่วนย่อย แล้วประมวลผลไปพร้อมกันได้หรือไม่ โดยอาจจะพิจารณาจากความสัมพันธ์ของชุดข้อมูล หรือเป็นกลุ่มของข้อมูลที่อยู่ติดกัน แล้วทำการประมวลผลโดยใช้หลักการ “โปรแกรมเดี่ยวหลายชุดข้อมูล” (Single Program Multiple Data: SPMD) ซึ่งโปรแกรมที่ทำงานอยู่บนแต่ละหน่วยประมวลผลจะเป็นโปรแกรมชุดเดียวกัน แต่ประมวลผลข้อมูลคนละชุดข้อมูล ซึ่งชุดข้อมูลที่ถูกส่งไปคำนวณจะเกิดจากการแบ่งปัญหาด้วยวิธีนี้ ตัวอย่างเช่น การบีบอัดข้อมูลภาพเคลื่อนไหวหรือภาพยนตร์ สามารถทำการแบ่งข้อมูลออกเป็นส่วน ๆ แล้วส่งไปประมวลผลยังแต่ละหน่วยประมวลผล เมื่อได้ผลลัพธ์จึงค่อยนำมารวมกันก็จะได้เป็นภาพยนตร์ที่ผ่านการเข้ารหัสที่สมบูรณ์ทั้งหมด ซึ่งจากตัวอย่างจะเห็นได้ว่า โปรแกรมที่ทำงานอยู่ในแต่ละหน่วยประมวลผลเป็นโปรแกรมเดียวกัน แต่ประมวลผลส่วนของข้อมูลต่างกัน ตามหลักการ SPMD
- การแบ่งปัญหาตามหน้าที่ของการคำนวณออกเป็นส่วนย่อย (Functional Decomposition) ซึ่งการแบ่งงานแบบนี้จะกระทำตั้งแต่เริ่มลงมือเขียนส่วนของโปรแกรม เพราะจะต้องทำการแยกฟังก์ชั่นการทำงานของแต่ละส่วนอย่างชัดเจนตั้งแต่ทีแรก ว่าส่วนไหนทำอะไร และเกี่ยวข้องกับข้อมูลชุดไหนบ้าง เพื่อให้เป็นไปตามหลักการ “หลายโปรแกรมหลายชุดข้อมูล” (Multiple Program Multiple Data: MPMD) เช่น การสร้างแบบจำลองของสภาพภูมิอากาศ ซึ่งประกอบด้วยงานย่อย ๆ คือ การสร้างแบบจำลองของทะเล พื้นดิน และชั้นบรรยากาศ



รูปที่ 2.5 การแบ่งขอบเขตหน้าที่ของปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ขั้นตอนออกแบบการสื่อสาร

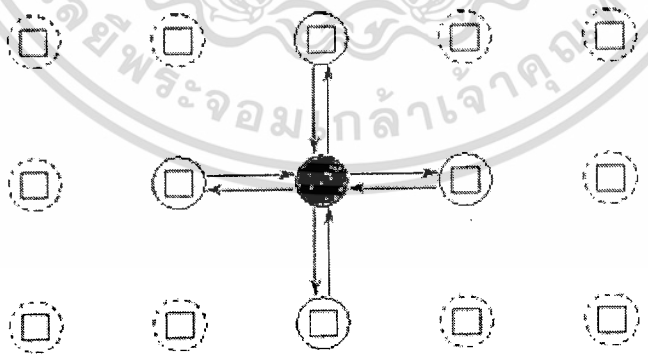
ขั้นตอนการสื่อสารนี้จะพิจารณาความสัมพันธ์ในเรื่องของการสื่อสารระหว่างงานที่แบ่งออกไปในงานบางอย่าง งานย่อยที่ถูกแบ่งไม่สามารถประมวลผลได้ในเวลาเดียวกันได้ทั้งหมด เนื่องจากต้องอาศัยข้อมูลหรือผลลัพธ์จากงานย่อยอื่นเพื่อใช้ในการประมวลผล ดังนั้นจึงต้องมีการสื่อสารระหว่างงานย่อย การเลือกวิธีการสื่อสารที่เหมาะสมจะทำให้การปะมวลผลมีประสิทธิภาพมากยิ่งขึ้น การออกแบบขั้นตอนการสื่อสารจะแบ่งเป็นสองขั้นตอน ขั้นตอนแรกออกแบบโครงสร้างของการเชื่อมโยงว่างานใดต้องการข้อมูลและงานใดทำหน้าที่ให้ข้อมูล ขั้นตอนที่สองคือกำหนดโครงสร้างของข้อความ (Message) ที่ใช้ในการส่งและรับของแต่ละการเชื่อมโยง ลักษณะของการสื่อสารอาจแบ่งได้เป็นสี่แบบดังนี้

- **การสื่อสารกับโปรเซสข้างเคียง (Local Process Communication)**

การสื่อสารกับโปรเซสข้างเคียง คือ การสื่อสารที่เกิดขึ้นระหว่างงานจำนวนน้อย เช่นการวิเคราะห์เชิงเลข (Numerical Analysis) ในวิธีไฟไนต์ดิฟเฟอเรนต์ของ Jacobi ซึ่งการหาค่า $X_{i,j}^{(t+1)}$ ทำได้จากสมการ 6.1 โดยใช้ข้อมูลห้าจุดในการคำนวณ

$$X_{i,j}^{(t+1)} = \frac{4X_{i,j}^{(t)} + X_{i-1,j}^{(t)} + X_{i+1,j}^{(t)} + X_{i,j-1}^{(t)} + X_{i,j+1}^{(t)}}{8} \tag{6.1}$$

งานแต่ละงานทำหน้าที่หาค่าของ $X_{i,j}^{(t+1)}$ ที่ตำแหน่งต่างๆกัน และในขณะเดียวกันก็ทำหน้าที่ส่งค่าของตัวเองให้กับงานอื่นด้วยดังรูปที่ 6.6



รูปที่ 2.6 ลักษณะการสื่อสารในการคำนวณไฟไนต์ดิฟเฟอเรนต์ในสองมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การสื่อสารแบบครอบคลุม (Global Communication)**

ส่วนการสื่อสารแบบครอบคลุมจะเป็นการสื่อสารในลักษณะที่งาน ๆ หนึ่งเป็นศูนย์กลางในการติดต่อกับงานอื่น ๆ ตัวอย่างเช่น การคำนวณหาค่าผลรวมดังสมการ 6.2

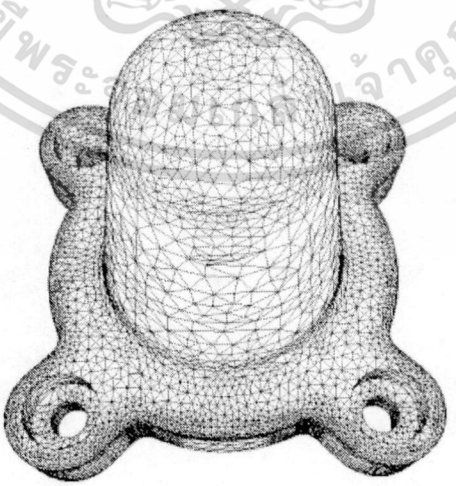
$$S = \sum_{i=0}^{N-1} X_i \tag{6.2}$$

การคำนวณขนานแบบนี้จะเกิดจากงานที่เป็นผู้จัดการ (Manager) รับค่า X_i ซึ่งเป็นผลลัพธ์จากการคำนวณของผู้ทำงาน (Worker) ต่าง ๆ เพื่อทำการหาค่าผลรวม S โดยการสื่อสารจะมีลักษณะดังรูปที่ 6.7

รูปที่ 2.7 การสื่อสารเพื่อรวมค่าโดยงานที่เป็นผู้จัดการ

- **การสื่อสารแบบไม่มีโครงสร้างและการสื่อสารแบบเปลี่ยนแปลงได้ (Unstructured and Dynamic Communication)**

ในหัวข้อที่ผ่านมาเป็นการสื่อสารแบบคงที่และมีโครงสร้างที่ไม่เปลี่ยนแปลง แต่มีงานบางรูปแบบที่มีโครงสร้างไม่แน่นอนและมีการเปลี่ยนแปลงตามเวลา เช่น การวิเคราะห์ไฟไนต์อีลีเมนต์ซึ่งแสดงดังรูปที่ 6.8



รูปที่ 2.8 ตาราง (Grid) ที่สร้างขึ้นเพื่อวิเคราะห์ไฟไนต์อีลีเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

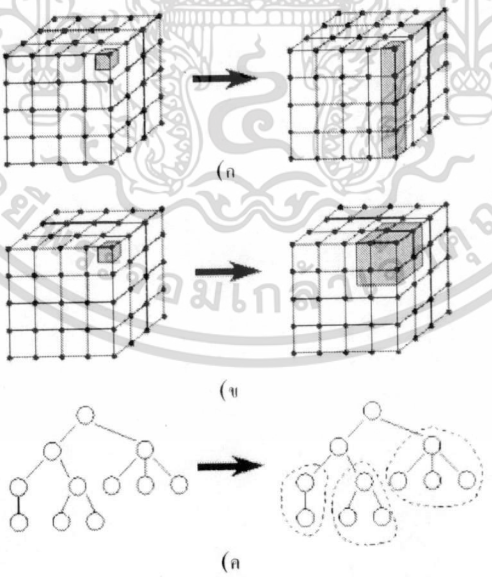
จากรูป ขนาดของตารางที่สร้างขึ้นมาเพื่อการวิเคราะห์ด้วยวิธีไฟไนต์อีลีเมนต์นี้จะไม่มีรูปแบบที่แน่นอน เพราะความละเอียดของตารางไม่เท่ากัน อาจจะขนาดใหญ่ในส่วนที่ไม่สำคัญมาก และจะมีขนาดเล็กลงมากขึ้นในส่วนที่เป็นจุดสนใจต้องการที่จะวิเคราะห์โดยละเอียด ดังนั้นจึงทำให้ไม่สามารถออกแบบการสื่อสารไว้ก่อนได้ วิธีแบ่งงานอาจทำได้โดยกำหนดให้แต่ละจุดต่อ (Vertex) ของเส้นตารางเป็นงานหนึ่งงาน และการสื่อสารจะทำตามจุดต่อตารางที่แบ่งไว้

- **การสื่อสารแบบไม่ประสานกัน (Asynchronous Communication)**

การสื่อสารทั้งสามรูปแบบที่กล่าวมาล้วนแต่เป็นการสื่อสารแบบประสาน หรือตรวจทานการทำงานทั้งสิ้น (Synchronous Communication) แต่จะมีการสื่อสารอีกรูปแบบหนึ่งที่ไม่มีการตรวจทานจังหวะและเวลาในการส่งข้อมูล ข้อมูลจะต้องถูกร้องขอออกมาก่อนเท่านั้น จึงจะเกิดการสื่อสารขึ้น การสื่อสารแบบประสานกัน ผู้ส่งและผู้รับรู้ว่าจะเกิดการสื่อสารกันก่อนจะเกิดการสื่อสารกันจริง ส่วนการสื่อสารแบบไม่ประสานกันผู้ส่งจะไม่รู้ว่าเมื่อใดผู้รับจะต้องการข้อมูล ผู้รับข้อมูลจะต้องทำการร้องขอข้อมูลโดยตรงจากผู้ส่งข้อมูลเมื่อต้องการ

2.4.3 ขั้นตอนการรวมกลุ่มงาน

ขั้นตอนรวมกลุ่มงานจะถูกทำเพื่อลดเวลาที่ใช้ในการสื่อสารระหว่างงานให้ลดลงและทำให้หน่วยประมวลผล (Processors) ไม่ต้องเสียเวลาในการสื่อสาร และรอคอยการรับส่งข้อมูล ทำให้ประสิทธิภาพในการคำนวณมากขึ้นด้วย



รูปที่ 2.9 การรวมกลุ่มงานในการประมวลผลแบบขนานเพื่อลดเวลาในการสื่อสาร

การรวมกลุ่มงานอาจทำได้หลายลักษณะขึ้นอยู่กับประเภทของงานเช่นจากรูปที่ 6.9

ก. เป็นการรวมกลุ่มงานจากงานในตารางสามมิติเป็นงานในตารางสองมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. เป็นการรวมกลุ่มงานจากงานในตารางสามมิติเป็นงานที่ใหญ่ขึ้นในสามมิติ

ค. เป็นการรวมกลุ่มงานในรูปแบบแผนภูมิต้นไม้เป็นต้น

2.4.4 ขั้นตอนการกำหนดงาน

ในขั้นตอนนี้จะเป็นการกำหนดงานไปยังหน่วยประมวลผลหรือหน่วยคำนวณที่เหมาะสม เพื่อเป็นการใช้ประโยชน์ของหน่วยประมวลผลสูงสุด และลดการสื่อสารให้มีน้อยที่สุด การกำหนดงานนี้สามารถทำเป็นแบบคงที่ที่กำหนดไว้ตายตัวตั้งแต่เริ่มสร้างโปรแกรม หรือทำการกำหนดงานตอนที่กำลังทำการคำนวณโดยใช้เทคนิคของการสมดุลงาน (Load Balancing) เข้ามาช่วยก็ได้

การกำหนดงานจะถูกทำเป็นขั้นตอนสุดท้ายในการออกแบบอัลกอริทึมแบบขนาน การกำหนดงานบนคอมพิวเตอร์แบบขนานจะต่างจากระบบคอมพิวเตอร์ที่มีหน่วยประมวลผลเดียว โดยในคอมพิวเตอร์ที่มีหน่วยประมวลผลเดียว ระบบปฏิบัติการจะเป็นผู้กำหนดและจัดลำดับงาน (Scheduling) ให้กับหน่วยประมวลผลเพื่อทำการประมวลผล ส่วนในคอมพิวเตอร์แบบขนานการกำหนดงานอาจทำโดยอัตโนมัติจากระบบปฏิบัติการแบบขนานเช่นในระบบคลัสเตอร์ที่ใช้คลัสเตอร์มิดเดิลแวร์อย่าง OpenMosix เป็นต้น หรืออาจกำหนดจากผู้สร้างโปรแกรมโดยตรง การกำหนดงานเป็นขั้นตอนที่ซับซ้อนซึ่งวิธีการกำหนดงานที่ดีจะทำให้ใช้เวลาในการประมวลผลงานน้อยและมีความยืดหยุ่นในการใช้งาน หลักการในการกำหนดงานที่ดีมีสองข้อคือ

- กำหนดงานที่สามารถประมวลผลพร้อมๆกันได้ให้ทำงานในหน่วยประมวลผลต่างกัน
- กำหนดงานที่ต้องมีการสื่อสารกันบ่อยให้ทำงานอยู่ในหน่วยประมวลผลเดียวกัน

การกำหนดงานไม่มีรูปแบบที่แน่นอนขึ้นอยู่กับประเภทของงาน รูปแบบของแต่ละปัญหาและลักษณะคอมพิวเตอร์แบบขนาน เช่น ในระบบคอมพิวเตอร์คลัสเตอร์แบบเนื้อผสม (Heterogeneous) ซึ่งความสามารถในการประมวลผลของแต่ละหน่วยประมวลผลอาจไม่เท่ากันจะทำให้วิธีการกำหนดงานมีความยากและซับซ้อนขึ้นไปอีก

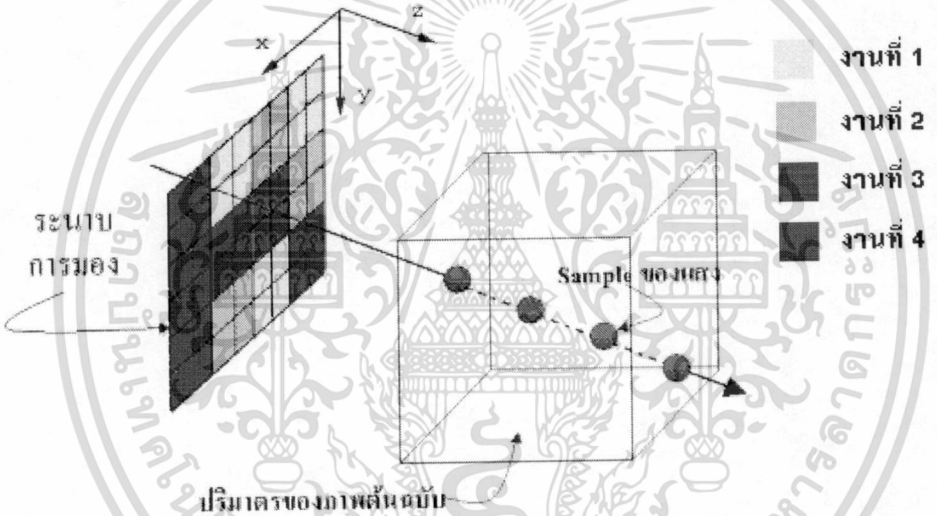
2.5 การสร้างภาพเชิงปริมาตรแบบฉายแสงบนระบบคลัสเตอร์

การสร้างภาพเชิงปริมาตรแบบฉายแสงบนระบบคลัสเตอร์ ได้ถูกนำเสนอไว้โดยนนท์ บัณฑิตวงษ์ (2545) เพื่อเป็นการเพิ่มความเร็วในการสร้างภาพเชิงปริมาตรในแต่ละมุมมองให้เร็วขึ้น ในวิธีการที่นำเสนอใช้หลักการที่ว่ากระบวนการสร้างภาพเชิงปริมาตรสามารถแบ่งการทำงานออกได้เป็นกลุ่มๆซึ่งไม่ขึ้นต่อกัน และไม่มีความจำเป็นต้องมีการสื่อสารระหว่างงานย่อย (Embarrassingly Parallel Computations) ซึ่งนับเป็นข้อได้เปรียบของงานประมวลผลภาพ และงานด้านคอมพิวเตอร์กราฟิคส่วนใหญ่ งานประเภทอื่นอาจต้องใช้วิธีการแบ่งงานที่ต่างออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 การแบ่งงานในการสร้างภาพเชิงปริมาตร

การแบ่งงานในการสร้างภาพเชิงปริมาตรใช้วิธีแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อย (Data Decomposition) โดยหน่วยของข้อมูลที่เล็กที่สุดคือจุดภาพ (Voxel) ตามแนวลำแสงดังแสดงในรูปที่ 6.10 การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อยมีข้อดีคือสามารถทำงานได้ไม่ว่าจะมีจำนวนโพรเซสอยู่ในระบบคลัสเตอร์เท่าใดก็ตาม (Scalable) แต่ต้องไม่มากกว่าจำนวนกลุ่มข้อมูลที่แบ่งไว้โดยสามารถแบ่งได้ละเอียดที่สุดถึงหนึ่งแนวลำแสงต่อหนึ่งงาน ปัญหาที่พบคือถ้าหากกำหนดให้ข้อมูลของปริมาตรภาพไปที่แต่ละโพรเซสในระบบคลัสเตอร์ เมื่อต้องการหมุนภาพจะต้องทำการกระจายชุดข้อมูลที่แบ่งนั้นไปให้แต่ละโพรเซสในระบบใหม่ ซึ่งข้อมูลภาพอาจมีขนาดใหญ่มากทำให้สิ้นเปลืองเวลาในการส่งข้อมูล วิธีการแก้ไขคือให้แต่ละโพรเซสทำการอ่านข้อมูลภาพทั้งปริมาตรไปเก็บไว้ในหน่วยความจำของตัวเอง เมื่อมีการหมุนภาพก็เพียงทำการหมุนตามขั้นตอนของการสร้างภาพเชิงปริมาตรดังที่กล่าวมาแล้วในบทที่ 3 แต่ข้อเสียของวิธีนี้คือทำให้สิ้นเปลืองหน่วยความจำ



รูปที่ 2.10 การแบ่งข้อมูลของปัญหาออกเป็นส่วนย่อยตามแนวลำแสง

2.5.2 การรวมกลุ่มงาน

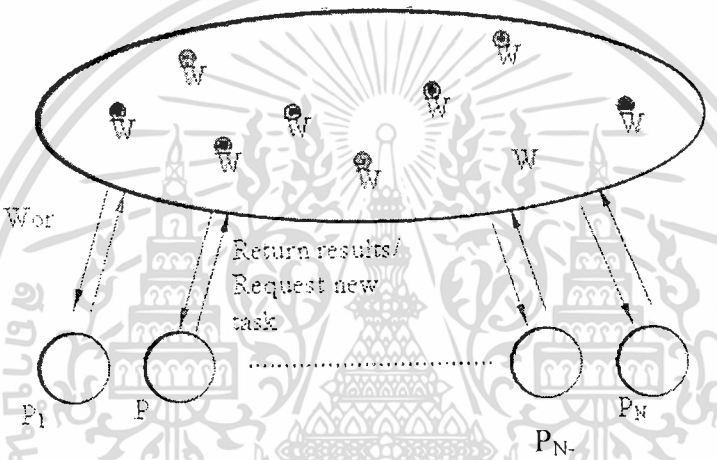
การรวมกลุ่มงานของการสร้างภาพเชิงปริมาตรนี้สามารถทำได้ง่ายโดยให้แนวลำแสงที่อยู่ติดกันรวมเป็นกลุ่มงานเดียวกัน การกำหนดให้งานหนึ่งงานเป็นการสร้างภาพในแนวหนึ่งลำแสงอาจเป็นงานที่ต้องอาศัยการคำนวณน้อยเกินไปและทำให้มีการสื่อสารมากเกินความจำเป็นซึ่งอาจทำให้เวลาที่ใช้ในการสร้างภาพส่วนใหญ่เสียไปกับการสื่อสารในขณะที่แต่ละโพรเซสใช้เวลาในการคำนวณของแต่ละงานน้อยมาก ตัวอย่างขนาดของกลุ่มงานที่สามารถใช้ได้ เช่น กลุ่มงานขนาด 32x32 หรือ 64x64 แนวลำแสง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 การสื่อสารระหว่างงานในการสร้างภาพเชิงปริมาตร

เนื่องจากงานที่ถูกแบ่งในการสร้างภาพเชิงปริมาตรไม่จำเป็นต้องมีการสื่อสารระหว่างกลุ่มงานด้วยกัน การออกแบบการสื่อสารจึงไม่มีความยุ่งยาก ซึ่งในรายการวิจัยฉบับนี้ใช้หลักการของศูนย์รวมงาน (Work Pool) ซึ่งเป็นการสื่อสารแบบสองทิศทางระหว่างโปรเซสที่เป็นผู้จัดการและโปรเซสที่เป็นผู้ทำงานโดยมีลักษณะดังรูปที่ 6.11 โครงสร้างของข้อมูลที่ใช้ส่งเป็นอินพุทให้แก่โปรเซสผู้ทำงานประกอบด้วยตำแหน่งของจุดเริ่มของบล็อกตามแนวแกน X และ Y ส่วนโครงสร้างของข้อมูลที่เป็นผลลัพธ์กลับมาจากโปรเซสผู้ทำงานจะประกอบด้วยตำแหน่งของจุดเริ่มของบล็อกที่ประมวลผลเสร็จและค่าสีของข้อมูลภาพภายในบล็อก

2.5.4 การกำหนดงานในการสร้างภาพเชิงปริมาตร



รูปที่ 2.11 การกำหนดงานแบบศูนย์รวมงาน (Work Pool)

การกำหนดงานแบบศูนย์รวมงานอาศัยหลักการที่ว่าให้โปรเซสที่สร้างขึ้นในระบบคลัสเตอร์ไปรับงานที่ต้องประมวลผลจากศูนย์รวมงาน แทนที่จะให้ผู้สร้างโปรแกรมเป็นผู้กำหนดงานที่ต้องทำให้แก่โปรเซสโดยตรงซึ่งทำให้ขาดความยืดหยุ่นในกรณีที่ต้องการให้มีจำนวนโปรเซสในระบบมากขึ้นหรือต้องการเปลี่ยนแปลงขนาดของกลุ่มงาน

2.6 สรุป

ในบทนี้ได้กล่าวถึงภาพกว้าง ๆ ของการโปรแกรมแบบขนาน โดยเริ่มจากรูปแบบหรือวิธีการในการพัฒนาโปรแกรม ว่าสามารถที่จะสร้างโปรแกรมได้จากวิธีไหนบ้าง แล้วตามด้วย เทคนิค,วิธีการที่ได้ถูกนำมาใช้ และปัญหาต่าง ๆ ที่ควรหลีกเลี่ยงในการสร้างโปรแกรม หลักจากนั้นเป็นการกล่าวถึงวิธีการในการออกแบบโปรแกรมแบบขนาน โดยได้นำเสนอขั้นตอนต่าง ๆ สี่ขั้นตอนดังนี้คือ ขั้นตอนการแบ่งงาน ขั้นตอนการสื่อสาร ขั้นตอนการรวมกลุ่มงาน และขั้นตอนการกำหนดงาน และในหัวข้อสุดท้ายของบทนี้ได้กล่าวถึงการสร้างภาพเชิงปริมาตรแบบฉายแสงบนระบบคอมพิวเตอร์คลัสเตอร์ ซึ่งได้อธิบายให้เห็นรายละเอียดต่าง ๆ โดยคร่าวแล้ว

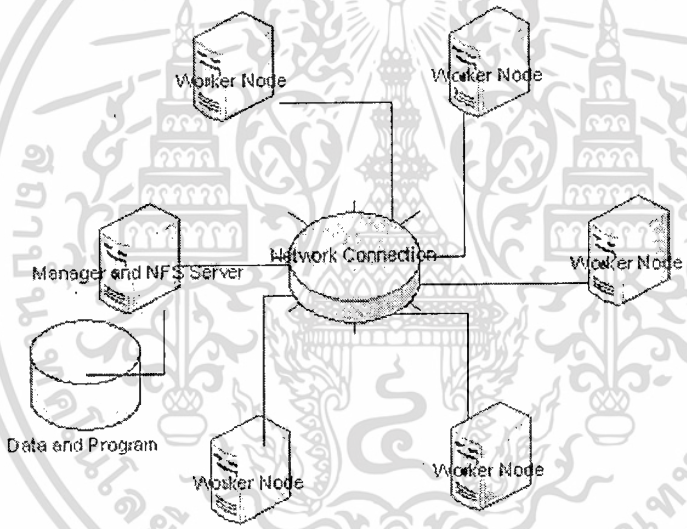
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ *3*

สรุปผลการวิจัย

3.1 โครงสร้างทาง Hardware

เครื่องให้บริการประมวลผลขนานความเร็วสูงที่จัดทำขึ้นในโครงการนี้ใช้รูปแบบการเชื่อต่อการทำงานระหว่างเครื่อง ตามรูปแบบที่เรียกว่า Computer Clustering Topology โดยโครงสร้างทางด้าน Hardware จะประกอบไปด้วย เครื่องคอมพิวเตอร์ที่เชื่อมต่อกันผ่านทาง LAN ความเร็ว 1000 Mbps (Gigabit Ethernet) มีการใช้งาน CPU, Memory (Distribute Memory) และ Disk แบบกระจาย แต่สามารถทำงานร่วมกันได้โดย การประสานการทำงานผ่านทาง Software Layer ซึ่งจะได้กล่าวต่อไป ดังนั้นจึงสามารถอธิบายโครงสร้างได้ดังรูปที่ 1 ดังนี้

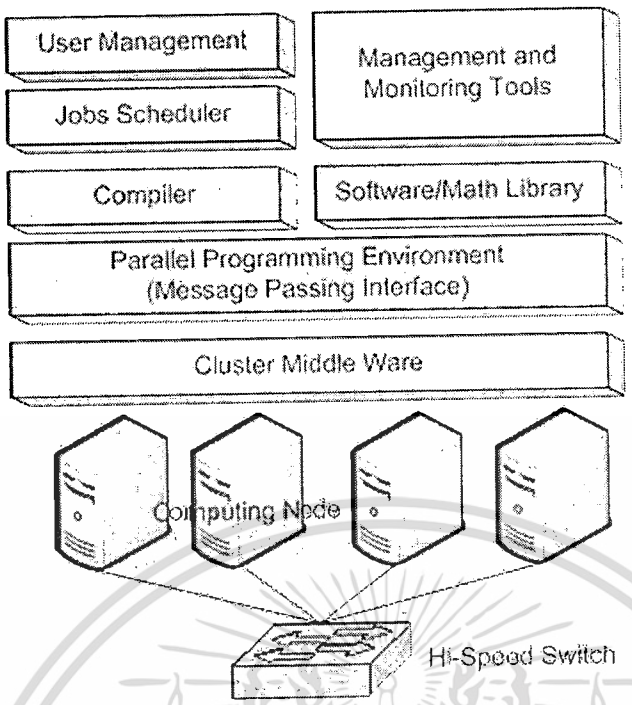


รูปที่ 3.1 โครงสร้างการเชื่อมต่อของคอมพิวเตอร์แบบคลัสเตอร์

3.2 โครงสร้างทาง Software

โครงสร้างทางด้าน Software จะประกอบด้วย Linux ซึ่งทำหน้าที่เป็นระบบปฏิบัติการและ MPICH Library ทำงานในส่วนของ Clustering Layer โดยจะทำงานในลักษณะที่เรียกว่า Parallel Programming Environment คือที่ทำหน้าที่เป็นตัวประสานการทำงานของคอมพิวเตอร์เครื่องต่าง ๆ ให้เกิดการประมวณผลแบบขนานขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงสร้างทางด้าน Software

3.3 รายการวัสดุอุปกรณ์

ตารางที่ 3.1 รายการวัสดุอุปกรณ์

ลำดับที่	รายการวัสดุ	จำนวน
1.	MainBoard: Tyan (Dual Socket)	6
2.	CPU: INTEL XEON 2.4GHZ-512K	12
3.	Hard Disk: 80 G	6
4.	RAM: ECC DDR PC2100 512 MB	12
5.	อุปกรณ์ที่จำเป็นอื่น ๆ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้